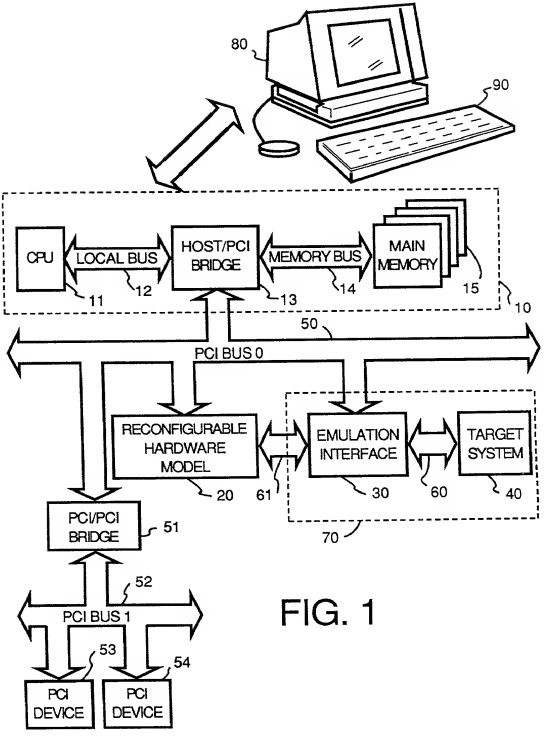


FIG. 1

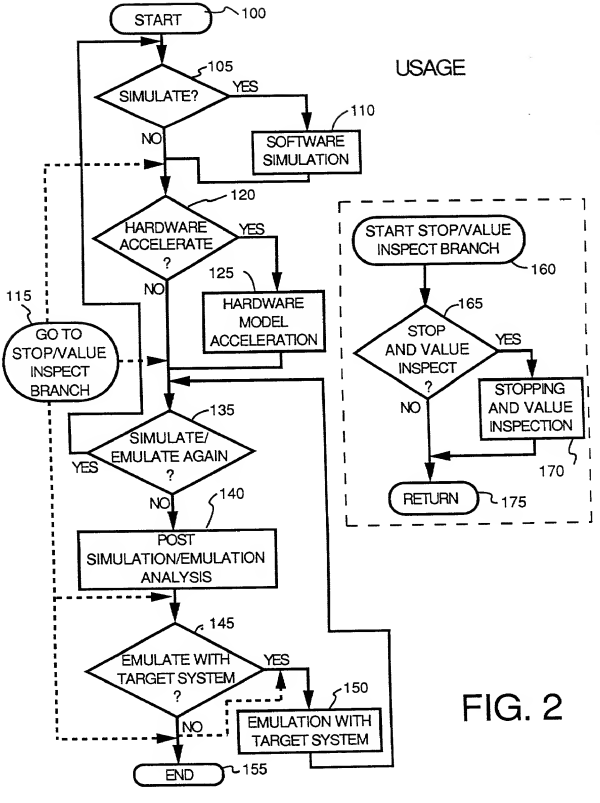

START
100
USAGE
105
SIMULATE?
YES
110
NO
SOFTWARE SIMULATION
120
HARDWARE ACCELERATE ?
YES
125
NO
HARDWARE MODEL ACCELERATION
115
GO TO STOP/VALUE INSPECT BRANCH
135
SIMULATE/ EMULATE AGAIN ?
YES
NO
140
POST SIMULATION/EMULATION ANALYSIS
145
EMULATE WITH TARGET SYSTEM ?
YES
150
NO
EMULATION WITH TARGET SYSTEM
END
155
START STOP/VALUE INSPECT BRANCH
160
165
STOP AND VALUE INSPECT ?
YES
NO
STOPPING AND VALUE INSPECTION
170
RETURN
175

FIG. 2

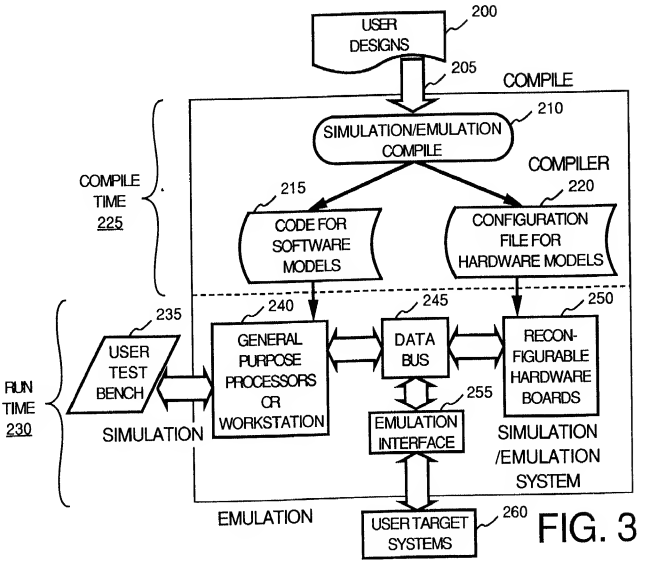


FIG. 3

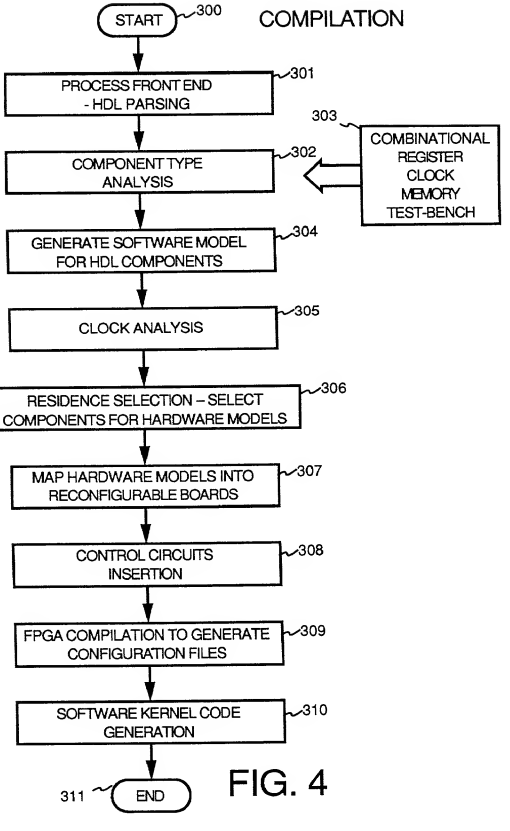

START
300
COMPILATION
PROCESS FRONT END
- HDL PARSING
301
303
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
COMPONENT TYPE
ANALYSIS
302
GENERATE SOFTWARE MODEL
FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION – SELECT
COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO
RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS
INSERTION
308
FPGA COMPILATION TO GENERATE
CONFIGURATION FILES
309
SOFTWARE KERNEL CODE
GENERATION
310
311
END

FIG. 4

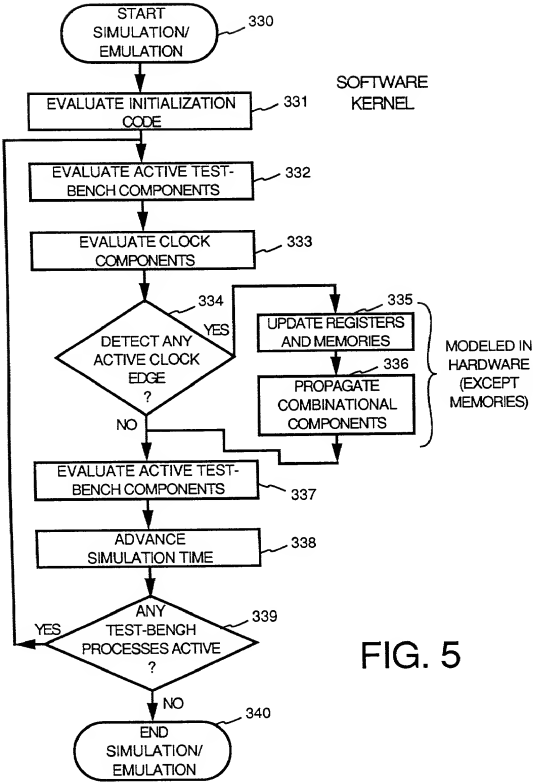

START SIMULATION/ EMULATION
330
SOFTWARE KERNEL
EVALUATE INITIALIZATION CODE
331
EVALUATE ACTIVE TEST-BENCH COMPONENTS
332
EVALUATE CLOCK COMPONENTS
333
334
DETECT ANY ACTIVE CLOCK EDGE ?
YES
NO
335
UPDATE REGISTERS AND MEMORIES
336
PROPAGATE COMBINATIONAL COMPONENTS
MODELED IN HARDWARE (EXCEPT MEMORIES)
EVALUATE ACTIVE TEST-BENCH COMPONENTS
337
ADVANCE SIMULATION TIME
338
ANY TEST-BENGH PROCESSES ACTIVE ?
339
YES
NO
END SIMULATION/ EMULATION
340

FIG. 5

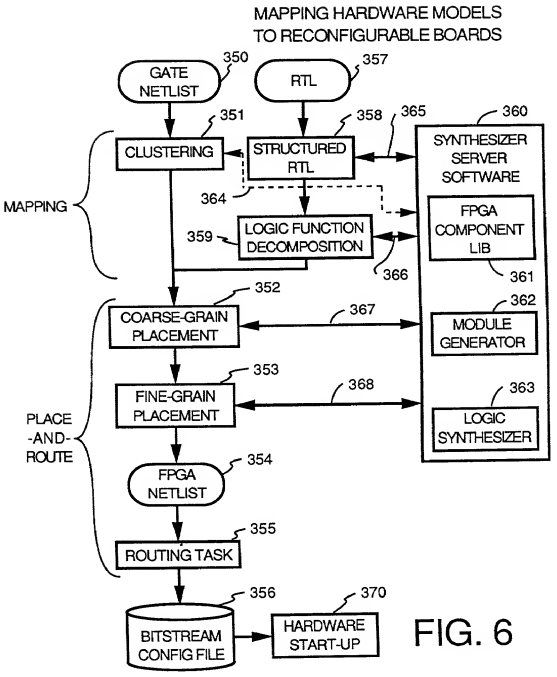


FIG. 6

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	1	0	0.	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	0	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

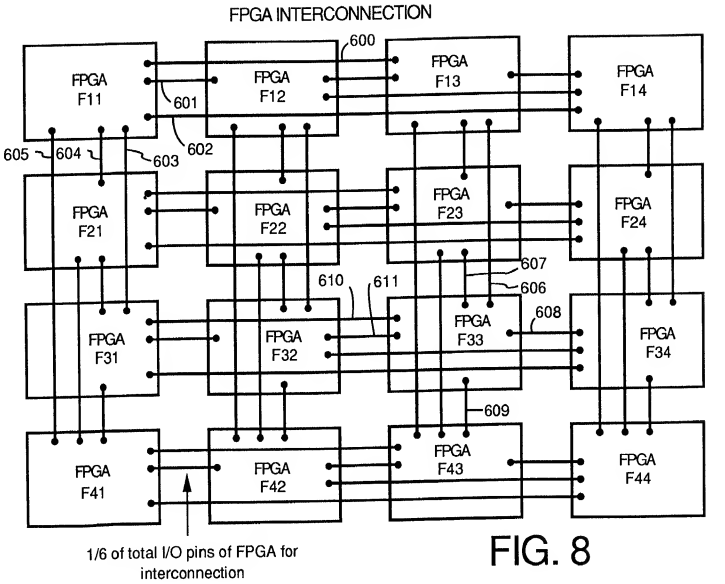


FIG. 8

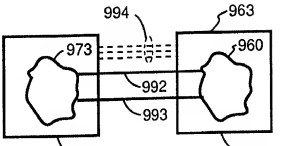


FIG. 9(a)

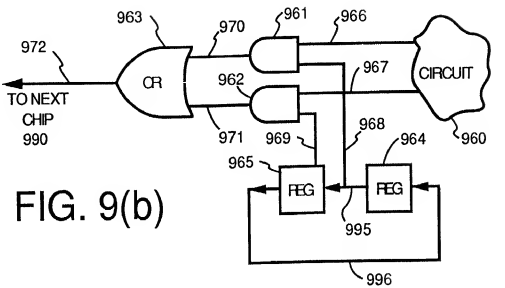


FIG. 9(b)

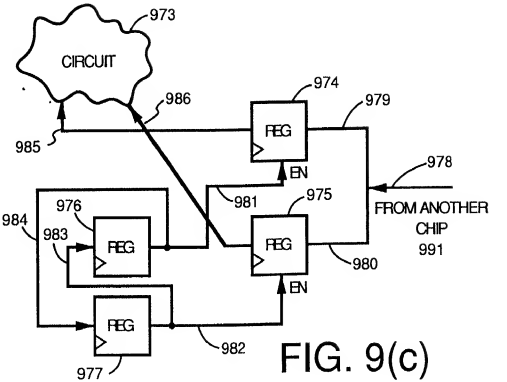


FIG. 9(c)

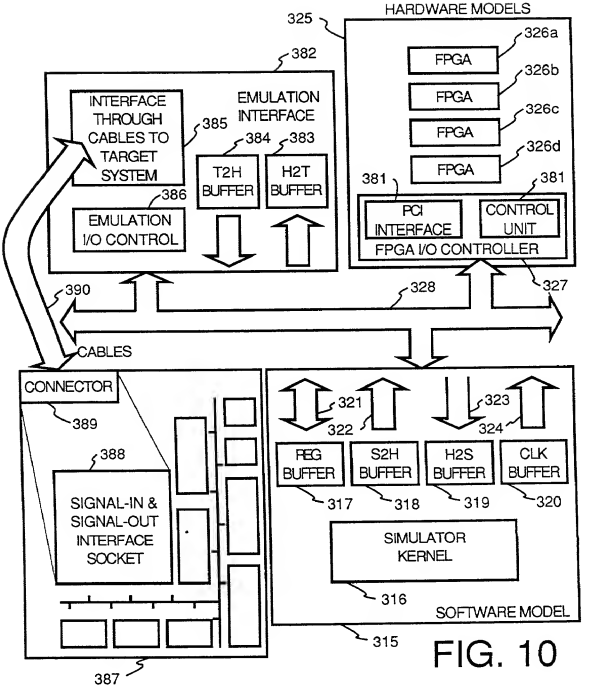


FIG. 10

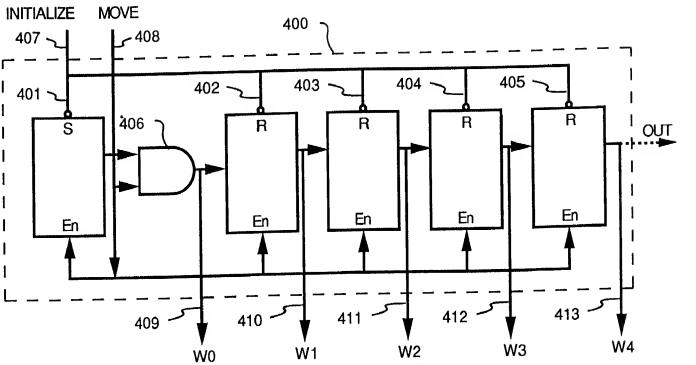


FIG. 11

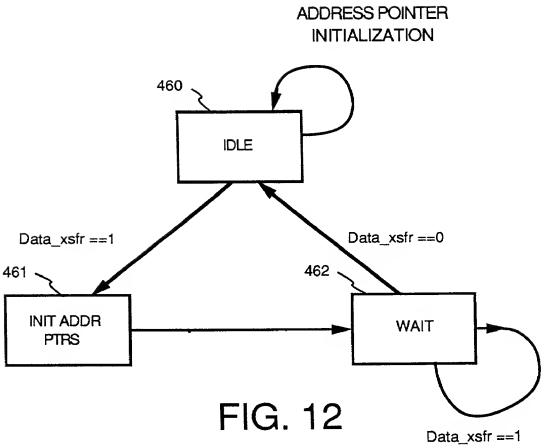


FIG. 12

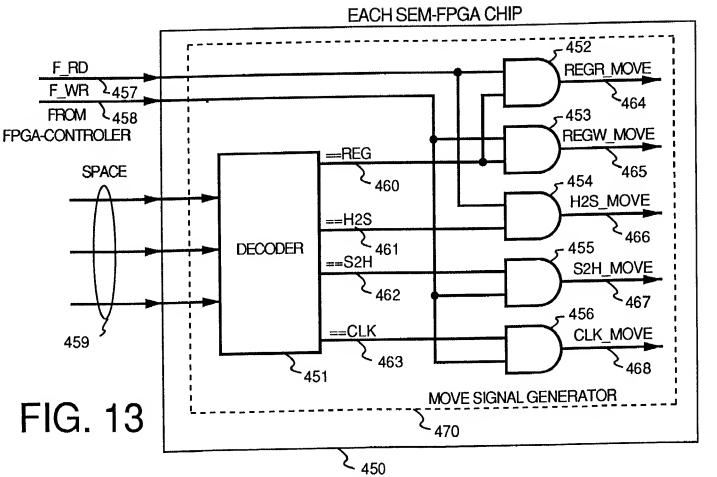


FIG. 13

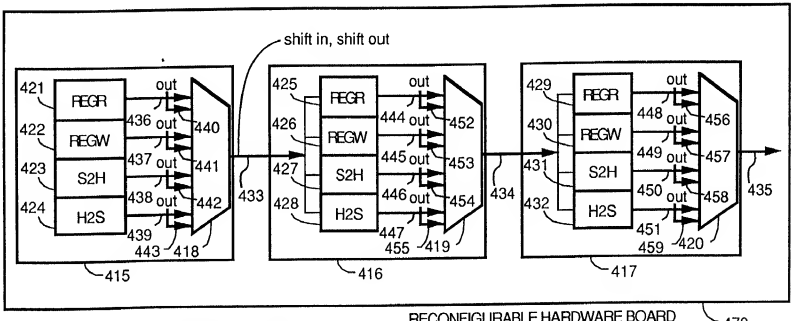


FIG. 14

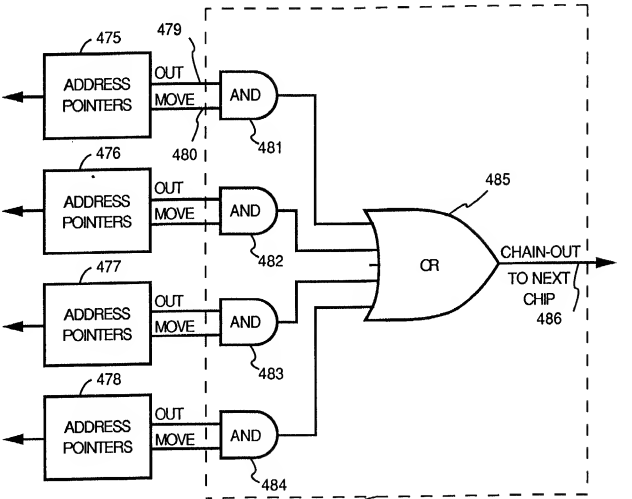


FIG. 15

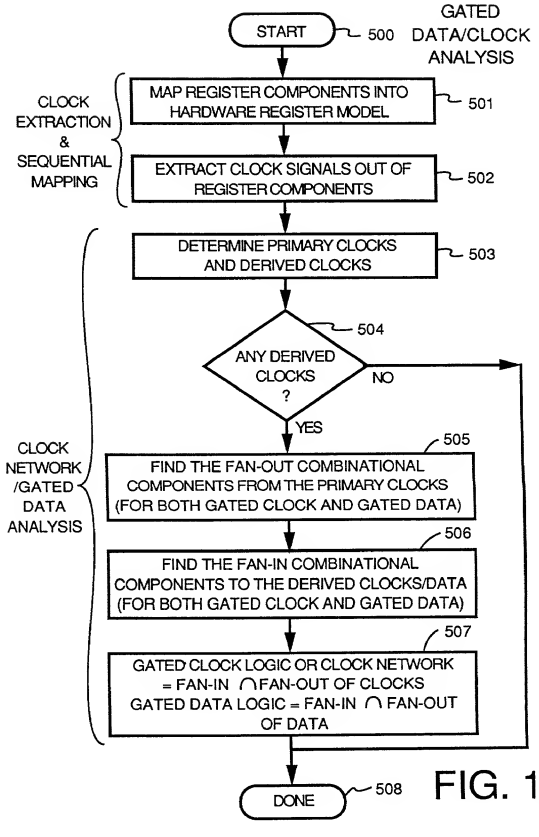

GATED DATA/CLOCK ANALYSIS
START
500
CLOCK EXTRACTION & SEQUENTIAL MAPPING
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
501
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
502
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
503
ANY DERIVED CLOCKS ?
504
NO
YES
CLOCK NETWORK /GATED DATA ANALYSIS
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
505
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
506
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN ∩ FAN-OUT OF DATA
507
DONE
508

FIG. 16

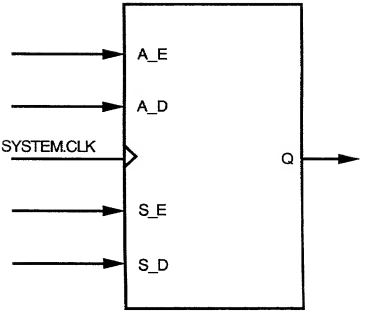


FIG. 17

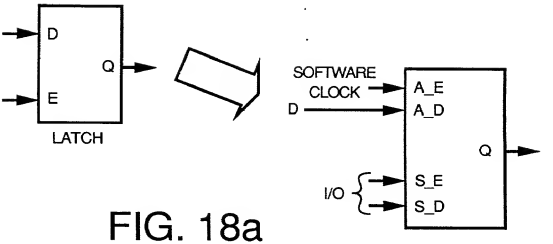


FIG. 18a

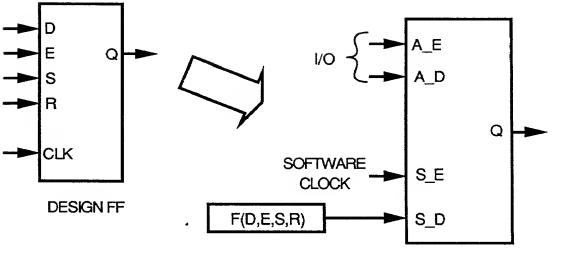


FIG. 18b

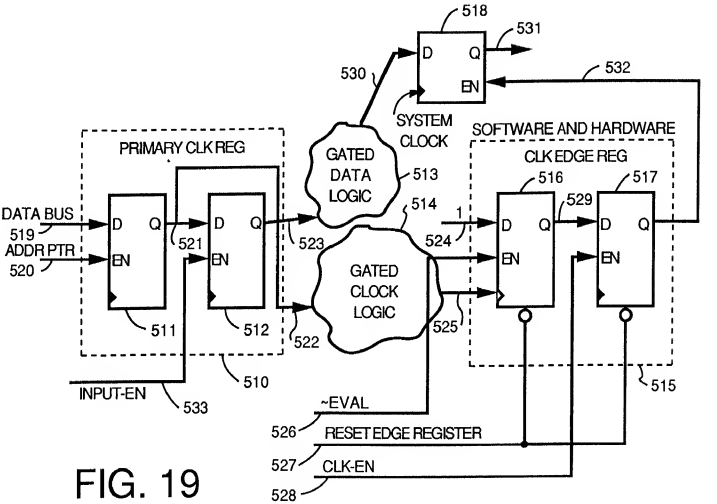


FIG. 19

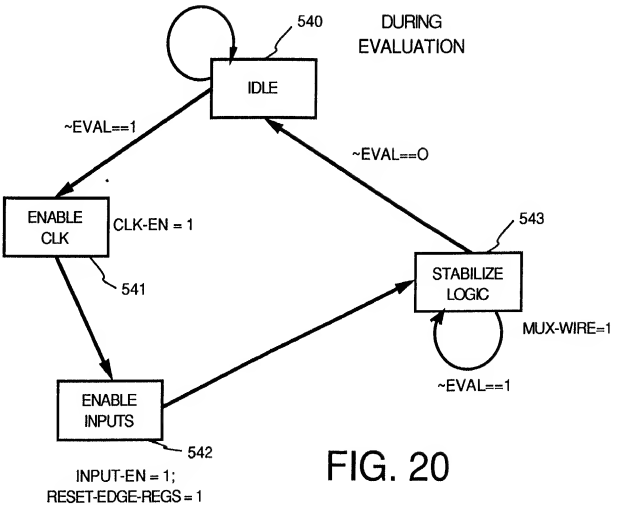


FIG. 20

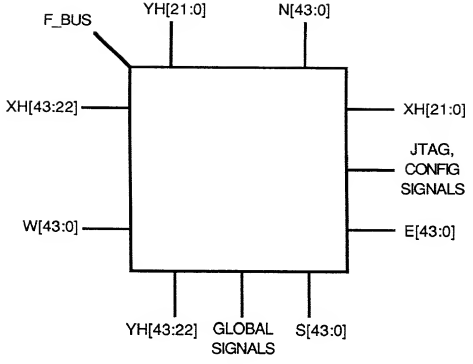


FIG. 21

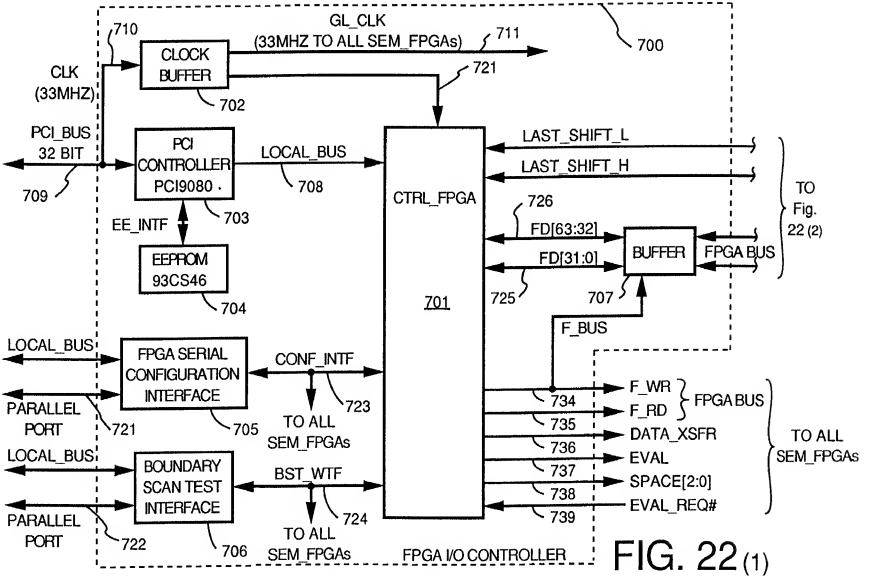


FIG. 22 (1)

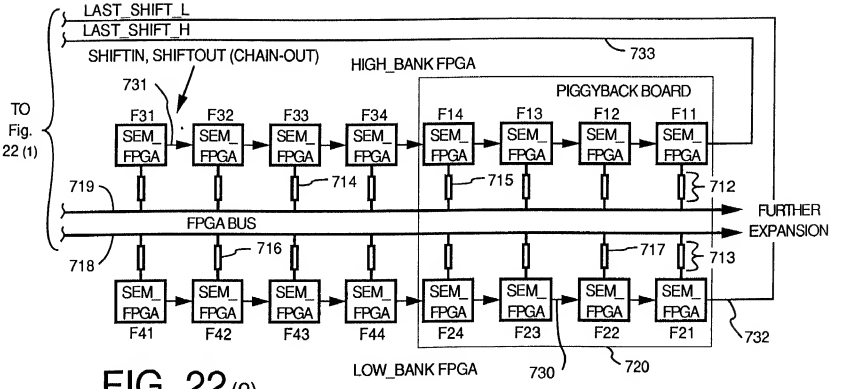


FIG. 22 (2)

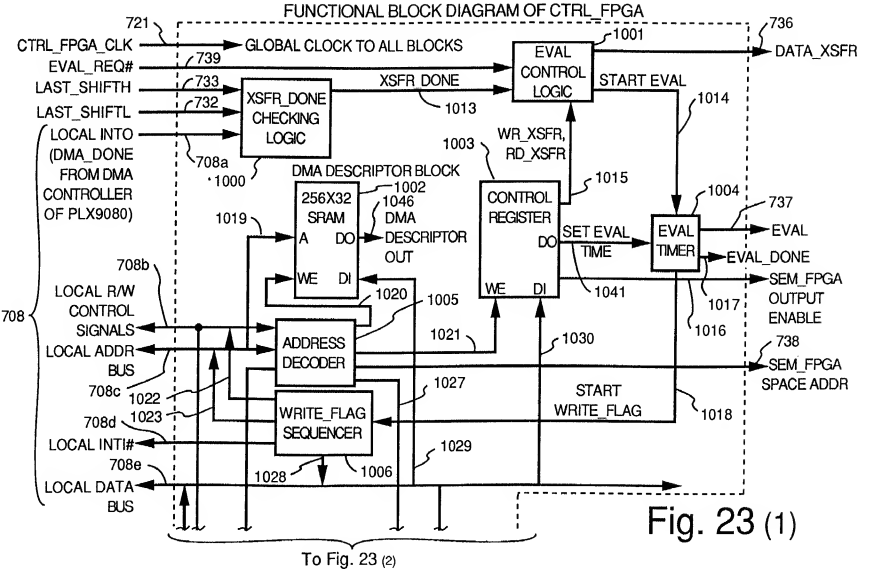


Fig. 23 (1)

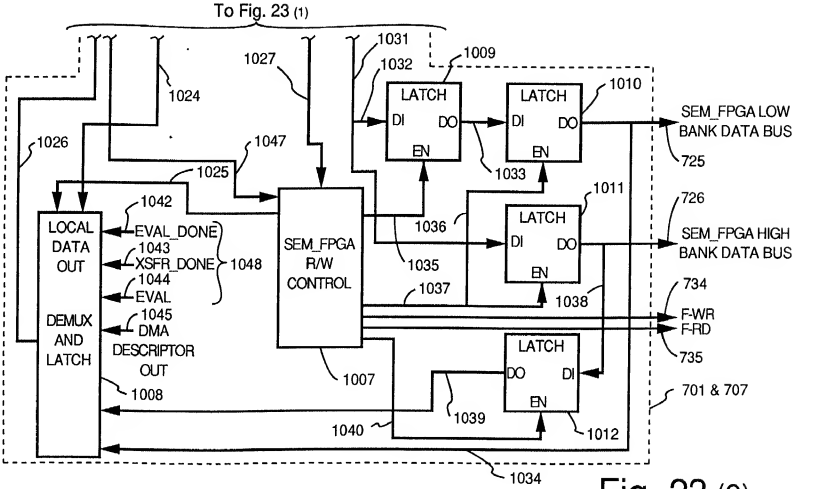


Fig. 23 (2)

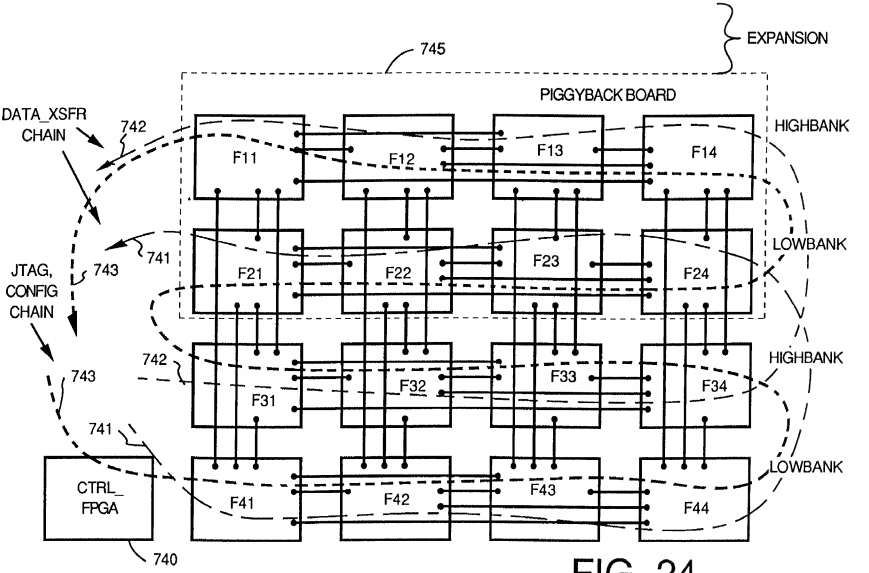


FIG. 24

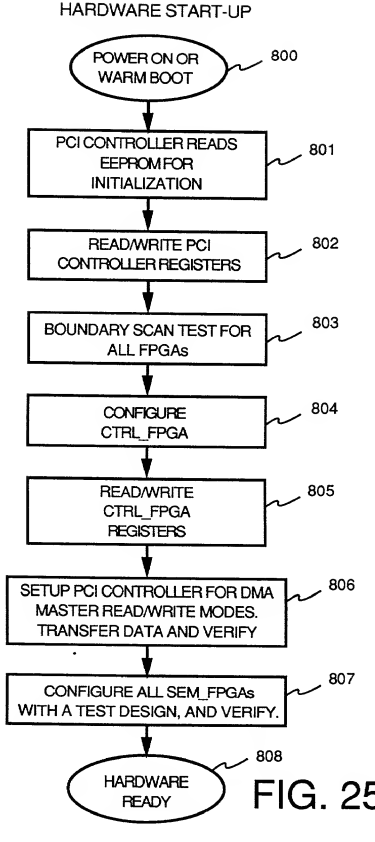


FIG. 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
   if(~reset)
      q = 0;
   else
      q = d;
endmodule

module example;
   wire d1, d2, d3;
   wire q1, q2, q3;
   reg sigin;
   wire sigout;
   reg clk, reset;

   register reg1 (clk, reset, d1, q1);
   register reg2 (clk, reset, d2, q2);
   register reg3 (clk, reset, d3, q3);

   assign d1 = sigin ^ q3;
   assign d2 = q1 ^ q3;
   assign d3 = q2 ^ q3;
   assign sigout = q3;

   // a clock generator
   always
   begin
      clk = 0;
      #5;
      clk = 1;
      #5;
   end

   // a signal generator
   always
   begin
      #10;
      sigin = $random;
   end

   // initialization
   initial
   begin
      reset = 0;
      sigin = 0;
      #1;
      reset =1;
      #5;
      $monitor($time, " %b, %b", sigin, sigout);
      #1000 $finish;
   end
   end module
```

Fig. 26

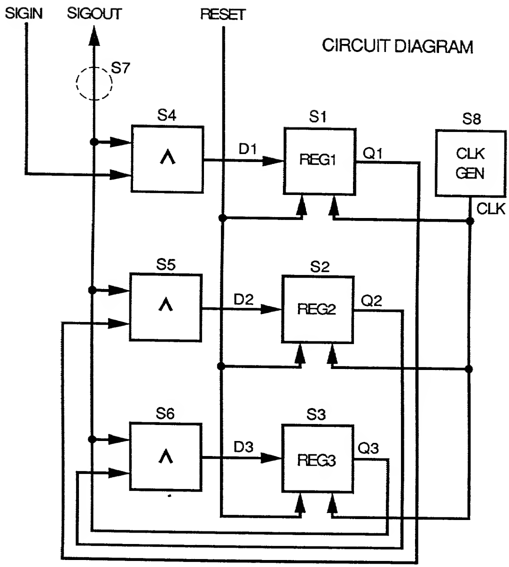

SIGIN
SIGOUT
RESET
CIRCUIT DIAGRAM
S7
S4
S1
S8
D1
REG1
Q1
CLK
GEN
CLK
S5
S2
D2
REG2
Q2
S6
S3
D3
REG3
Q3

FIG. 27

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(post edge clock or negedge reset)
   if(~reset)
     q = 0
   else
     q = d;

endmodule
```

Register definition

900

```
module example;
   wire d1, d2, d3;
   ware q1, q2, q3;
```

Wire interconnection info

907

```
   reg sigin;
   wire sigout;
   reg clk, reset;
```

Test-bench input -- 908

Test-bench output -- 909

```
S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);
```

Register component

901

```
S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;
```

Combinational component

902

```
S8  // a clock generator
    always
    begin
      clk = 0;
      #5;
      clk = 1;
      #5;
    end
```

Clock component

903

```
S9  // a signal generator
    always
    begin
      #10;
      sigin = $random;
    end
```

Test-bench component (Driver)

904

Fig. 28

```
    // initialization
    initial
    begin
S10   reset = 0;
      sigin = 0;
      #1;
S11   reset = 1;
      #5;
S12   $monitor($time, "%b, %b", sigin, sigout);
      #1000 $finish;
    end
    end module
```

Test-bench component (initialization)

905

Test-bench component (monitor)

906

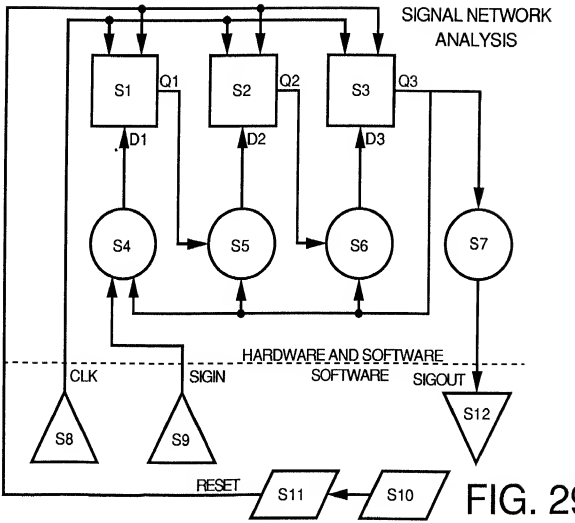


FIG. 29

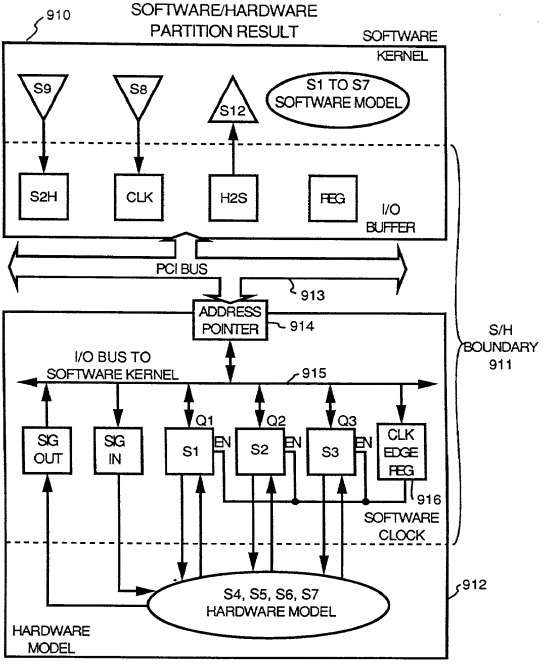

910
SOFTWARE/HARDWARE
PARTITION RESULT
SOFTWARE
KERNEL
S9
S8
S12
S1 TO S7
SOFTWARE MODEL
S2H
CLK
H2S
REG
I/O
BUFFER
PCI BUS
913
ADDRESS
POINTER
914
S/H
BOUNDARY
911
I/O BUS TO
SOFTWARE KERNEL
915
Q1
Q2
Q3
SIG
OUT
SIG
IN
S1
EN
S2
EN
S3
EN
CLK
EDGE
REG
916
SOFTWARE
CLOCK
912
S4, S5, S6, S7
HARDWARE MODEL
HARDWARE
MODEL

FIG. 30

HARDWARE MODEL

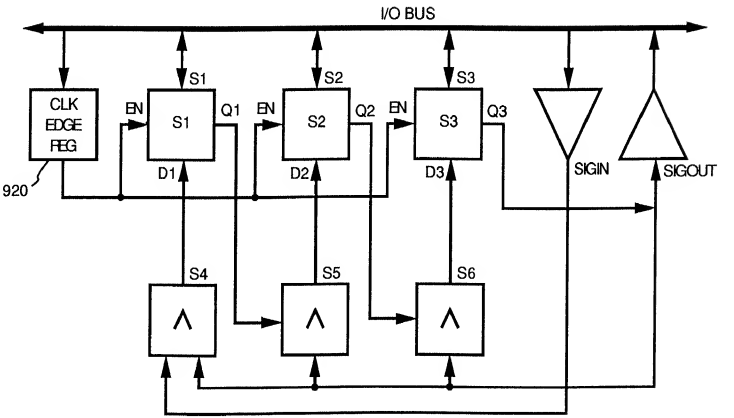


FIG. 31

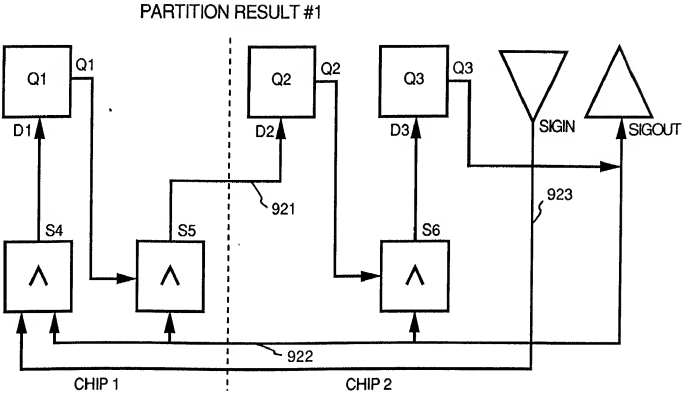


FIG. 32

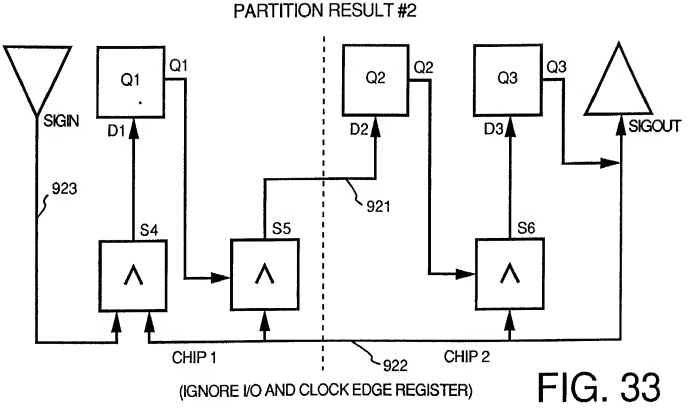


FIG. 33

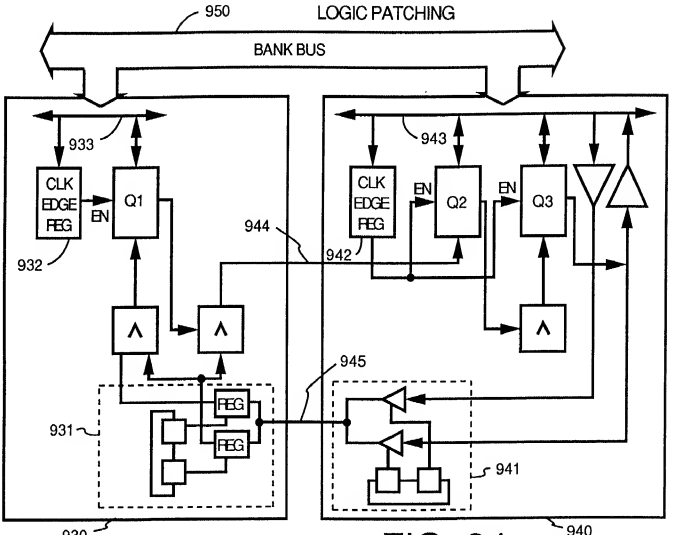


FIG. 34

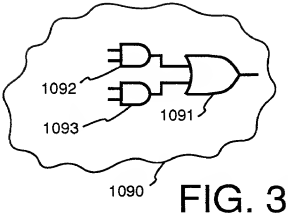


FIG. 35a

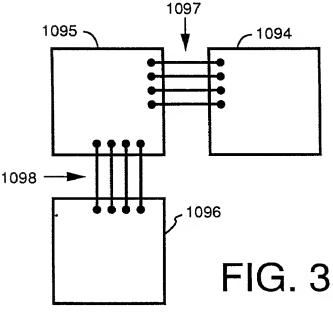


FIG. 35b

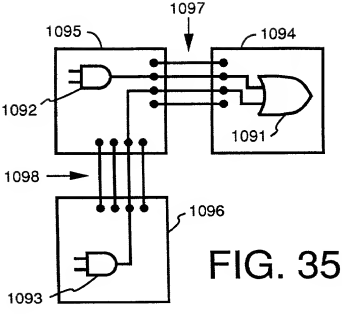


FIG. 35c

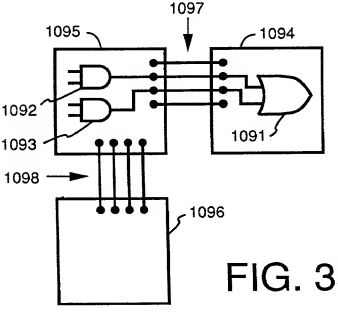
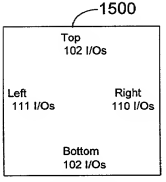


FIG. 35d

I/O PIN OVERVIEW OF FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V with 599-pin PGA package



425 Interconnect I/O pins

45 Dedicated I/O pins:

GCLK, FD_BUS[31..0], F_RD, F_WR, DATAXSFR, SHIFTIN, SHIFTOUT, SPACE[2..0], EVAL, EV_REQ_N, DEV_OE, DEV_CLRN

FIG. 36

FPGA INTERCONNECT BUSES

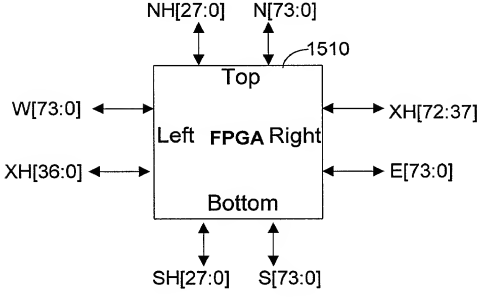


FIG. 37

BOARD CONNECTION - SIDE VIEW

DUAL-BOARD CONFIGURATION

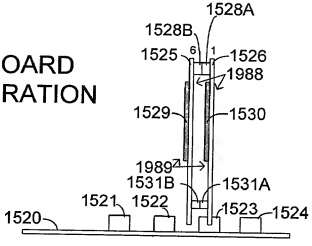


FIG. 38(A)

SIX BOARD CONFIGURATION

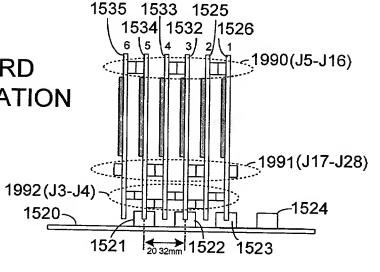


FIG. 38(B)

SIX-BOARD CONFIGURATION DIRECT-NEIGHBOR
AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

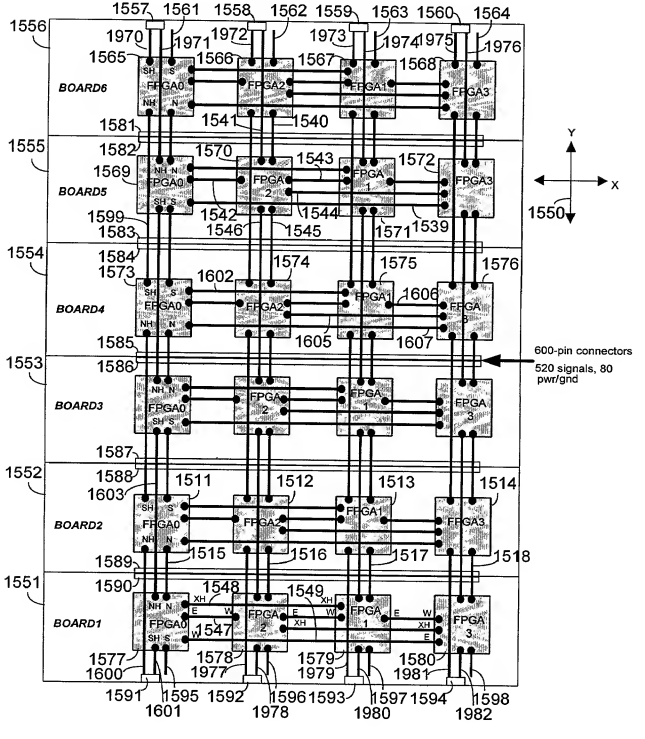


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

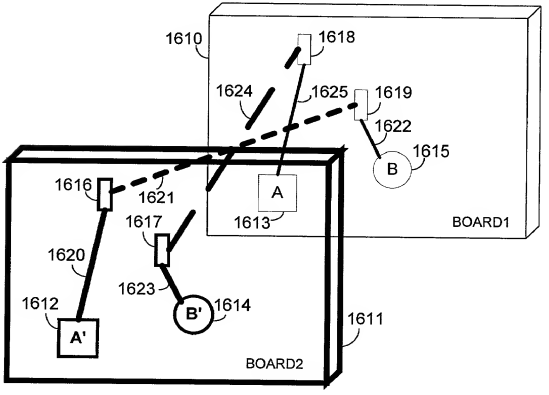


FIG. 40(A)

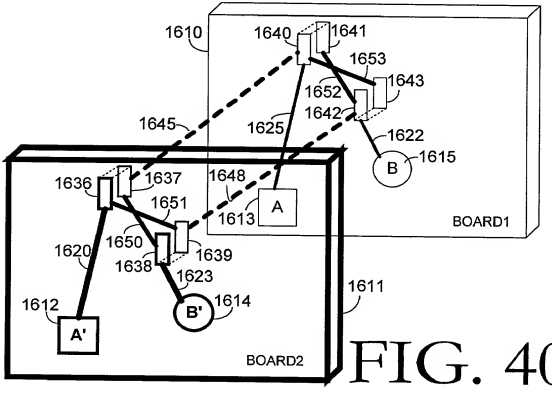


FIG. 40(B)

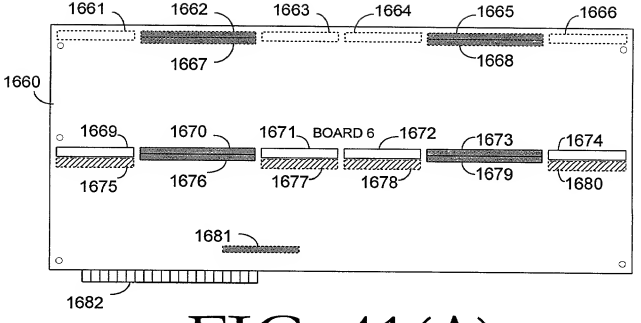


FIG. 41(A)

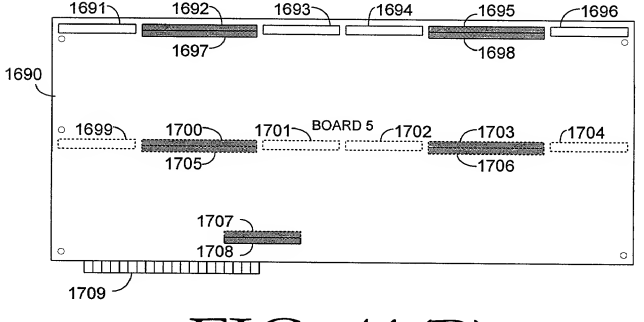


FIG. 41(B)

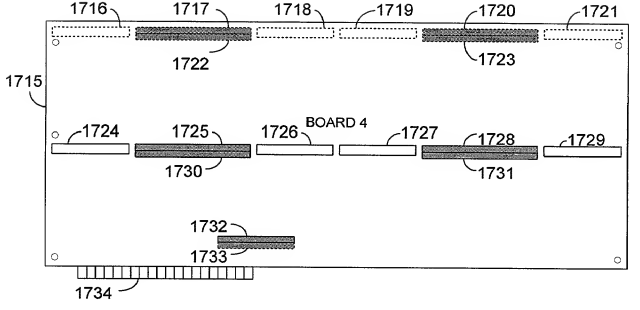


FIG. 41(C)

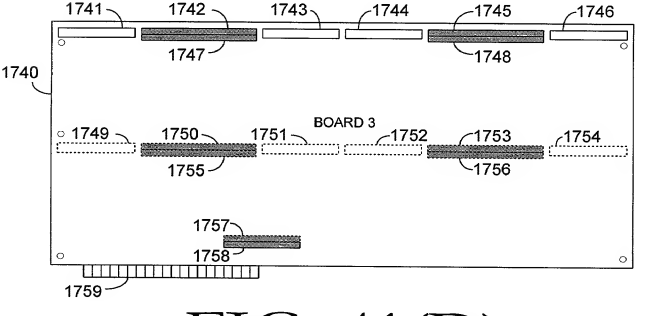


FIG. 41(D)

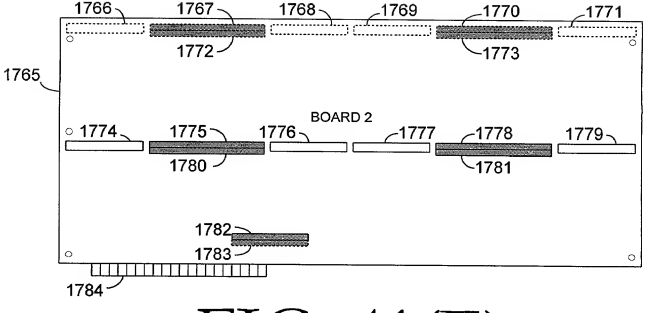


FIG. 41(E)

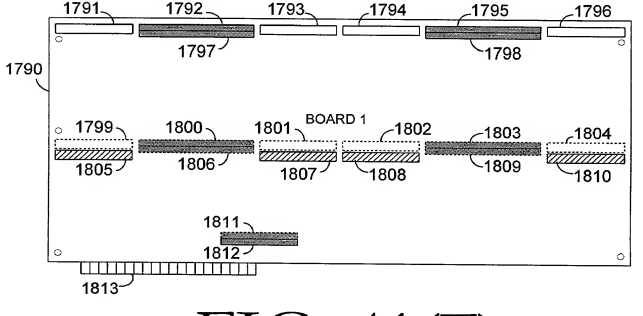


FIG. 41(F)

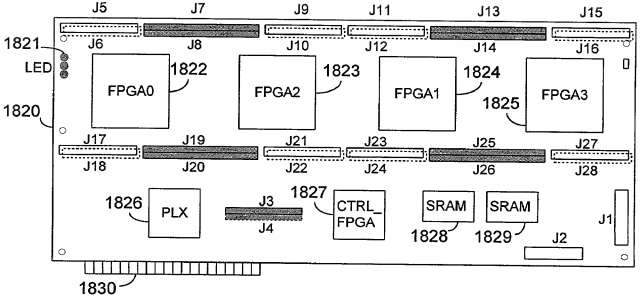


FIG. 42

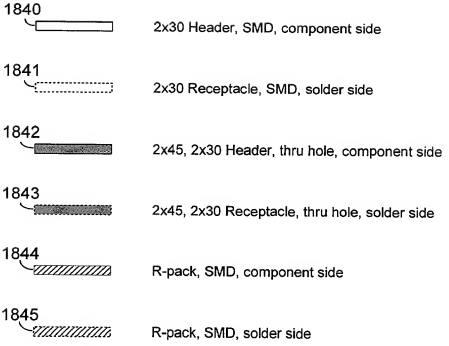


FIG. 43

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

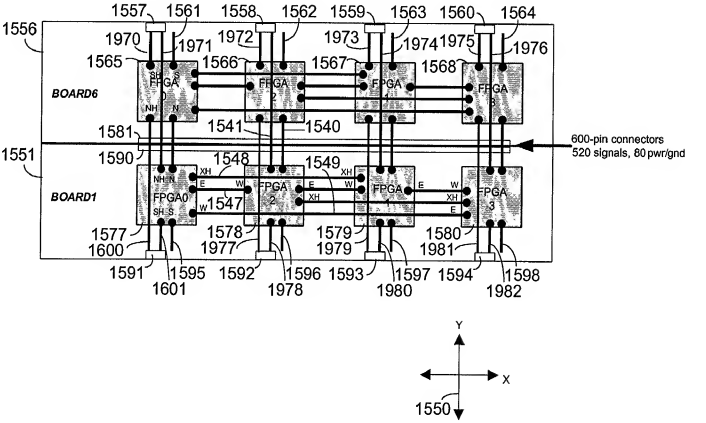


FIG. 44

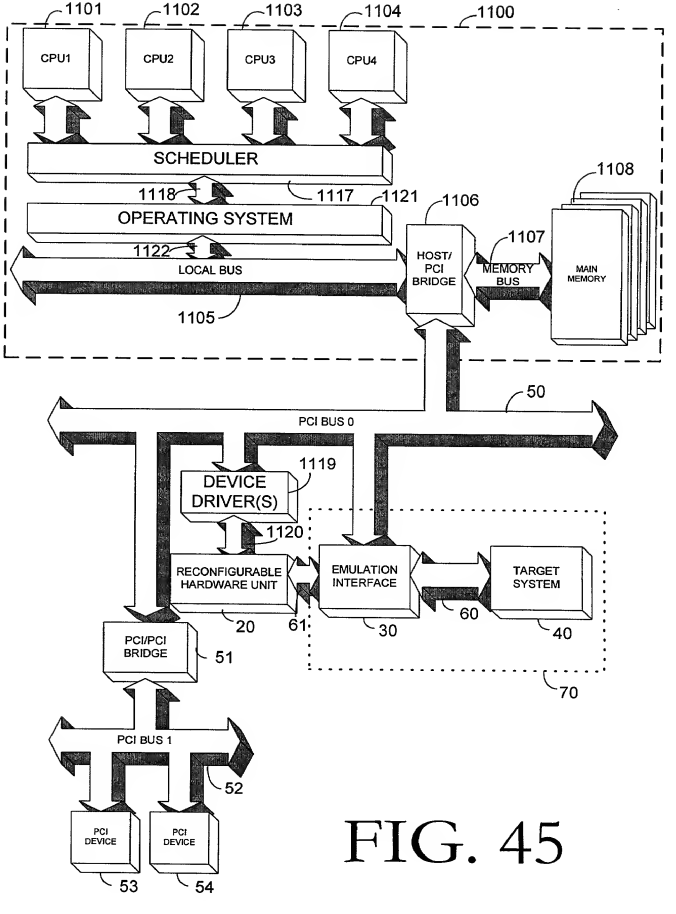
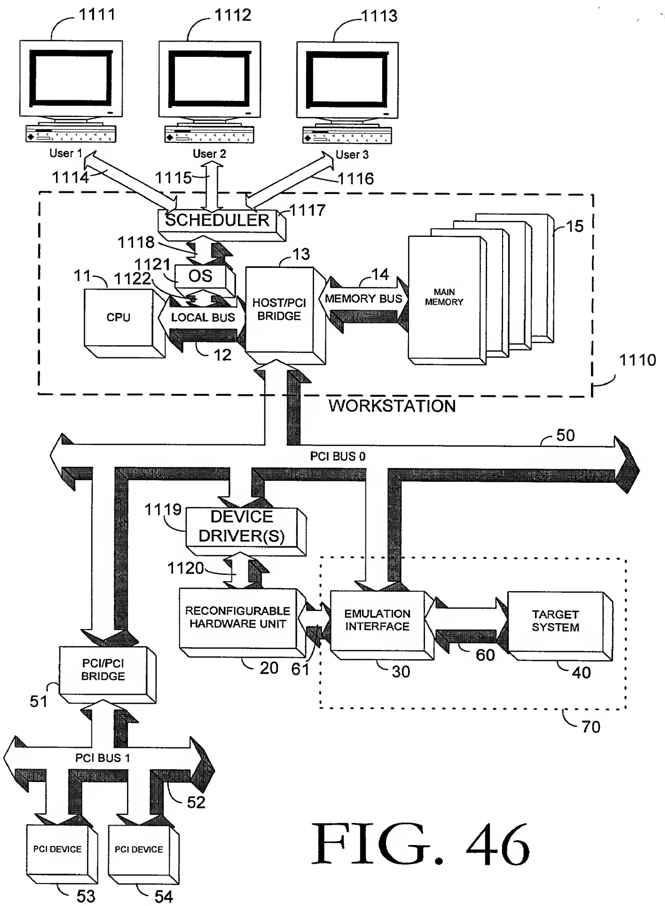


FIG. 45



1111
1112
1113
User 1
User 2
User 3
1114
1115
1116
1117
SCHEDULER
1118
1121
OS
1122
11
CPU
LOCAL BUS
12
13
HOST/PCI BRIDGE
14
MEMORY BUS
MAIN MEMORY
15
1110
WORKSTATION
50
PCI BUS 0
1119
DEVICE DRIVER(S)
1120
RECONFIGURABLE HARDWARE UNIT
20
61
EMULATION INTERFACE
30
60
TARGET SYSTEM
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 46

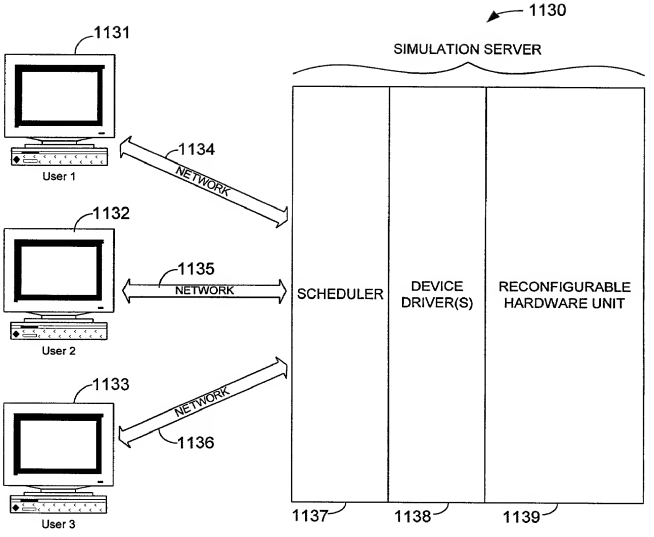

1130
SIMULATION SERVER
1131
User 1
1134
NETWORK
1132
User 2
1135
NETWORK
1133
NETWORK
1136
User 3
SCHEDULER
DEVICE DRIVER(S)
RECONFIGURABLE HARDWARE UNIT
1137
1138
1139

FIG. 47

SIMULATION SERVER ARCHITECTURE

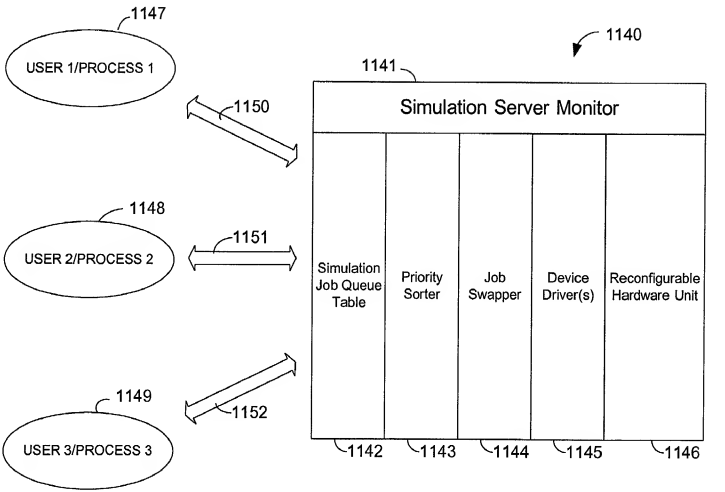


FIG. 48

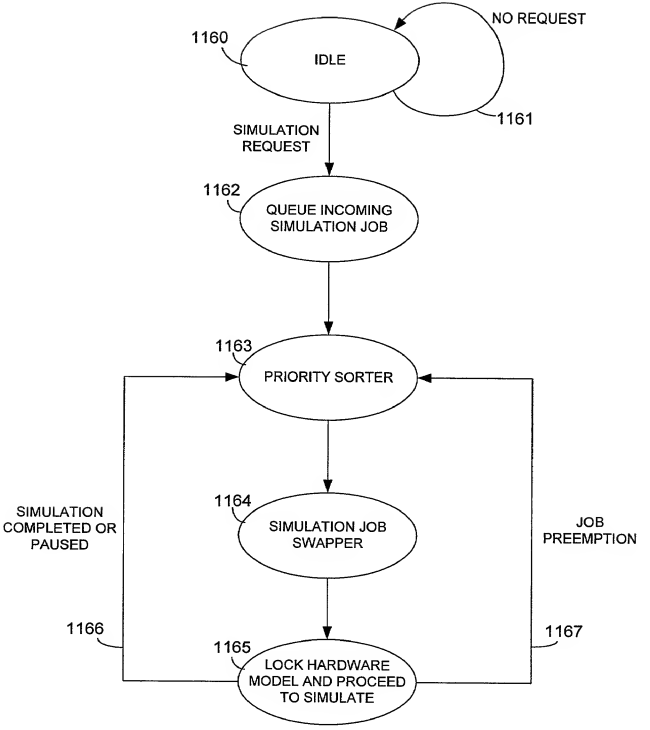


FIG. 49

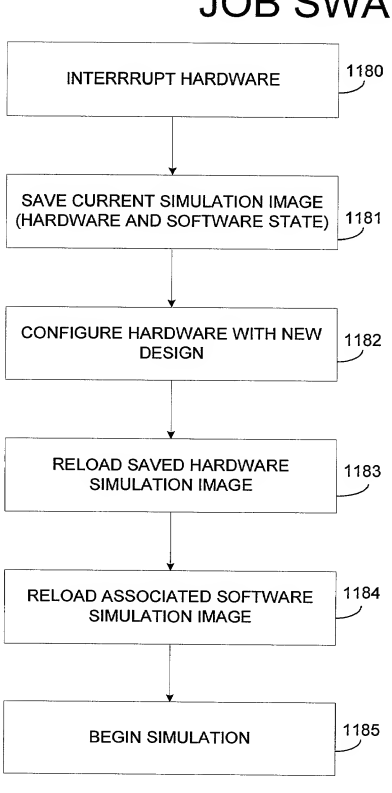


FIG. 50

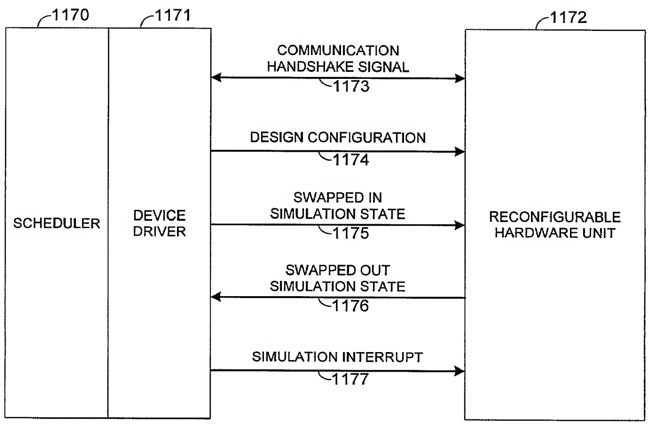

1170
1171
1172
SCHEDULER
DEVICE DRIVER
RECONFIGURABLE HARDWARE UNIT
COMMUNICATION HANDSHAKE SIGNAL
1173
DESIGN CONFIGURATION
1174
SWAPPED IN SIMULATION STATE
1175
SWAPPED OUT SIMULATION STATE
1176
SIMULATION INTERRUPT
1177

FIG. 51

PRIORITY I { JOB A, JOB B

PRIORITY II { JOB C, JOB D

TIME-SHARED HARDWARE USAGE:

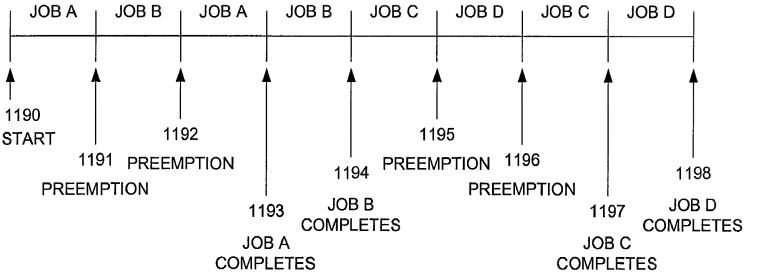


FIG. 52

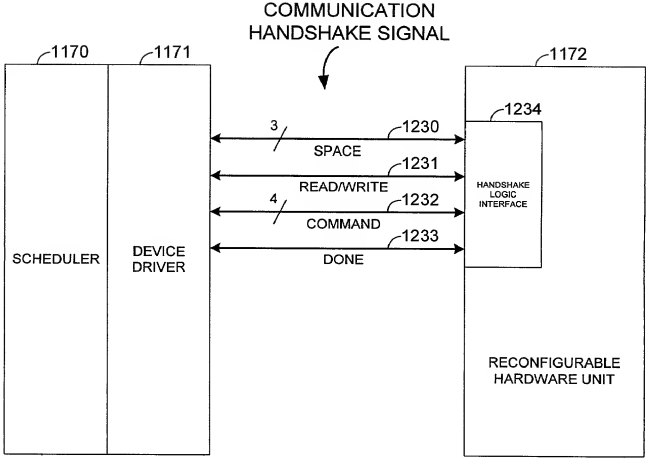

COMMUNICATION
HANDSHAKE SIGNAL
1170
1171
1172
1234
3
1230
SPACE
1231
READ/WRITE
4
1232
COMMAND
1233
DONE
HANDSHAKE
LOGIC
INTERFACE
SCHEDULER
DEVICE
DRIVER
RECONFIGURABLE
HARDWARE UNIT

FIG. 53

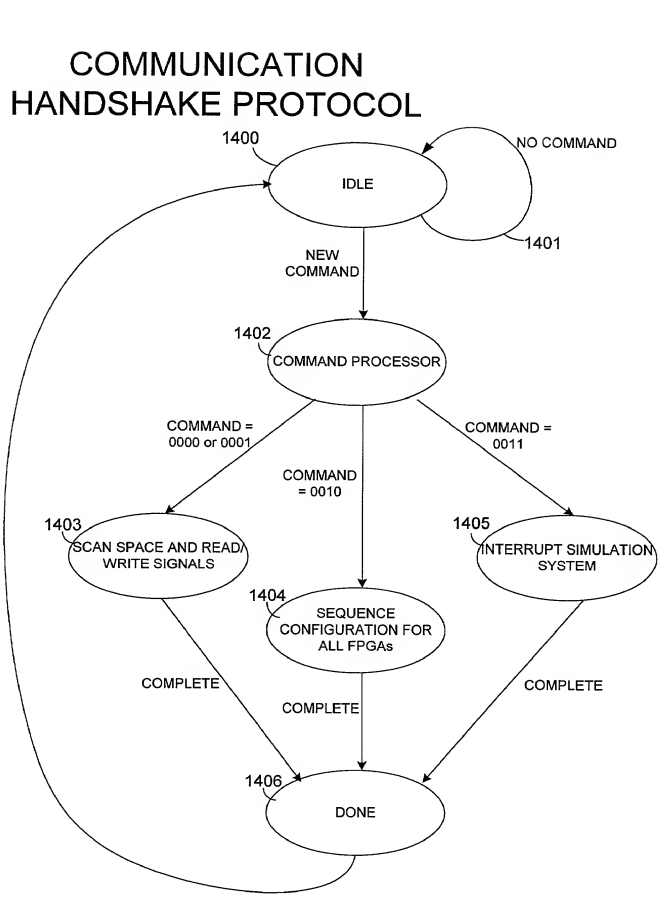


FIG. 54

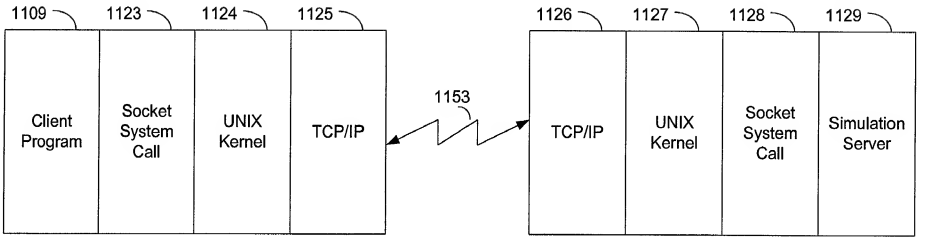


FIG. 55

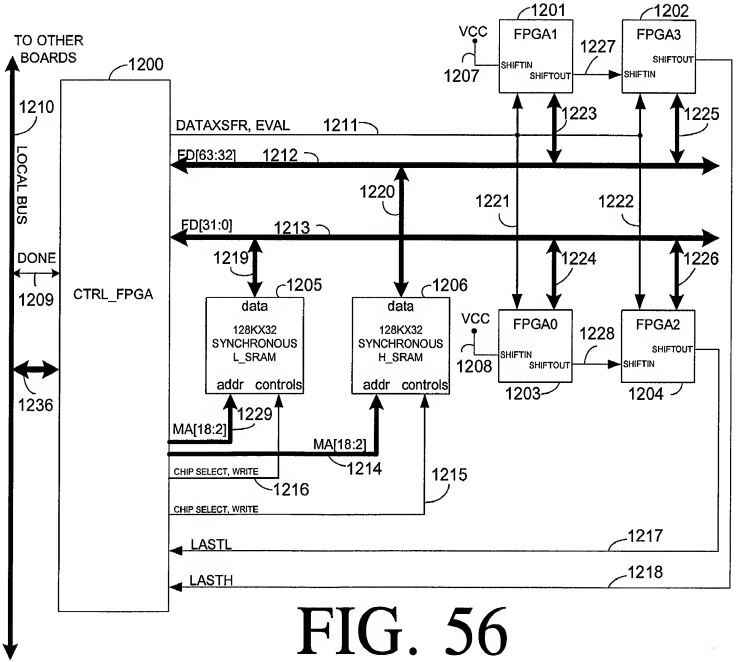


FIG. 56

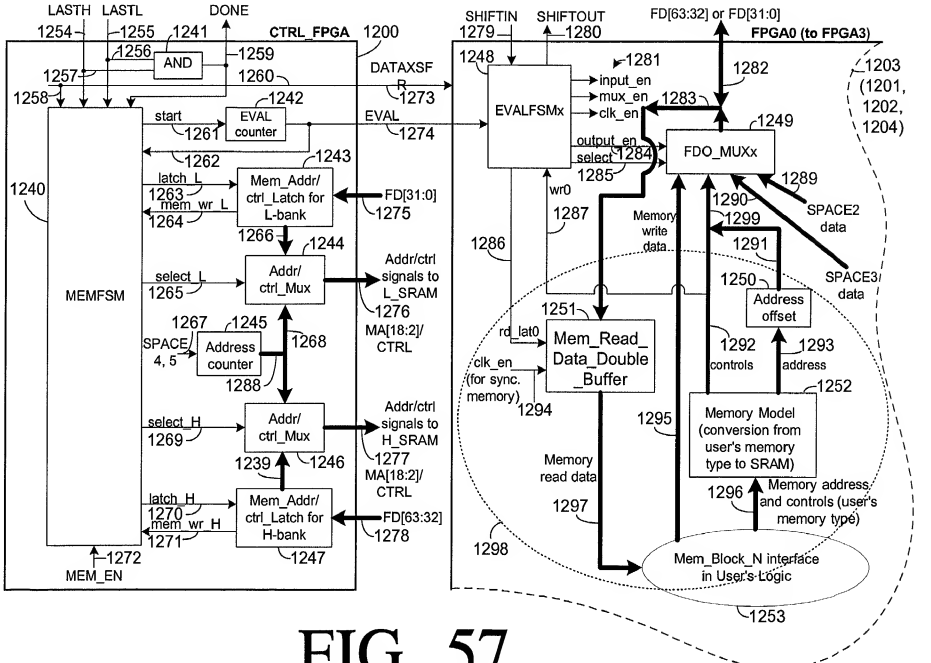


FIG. 57

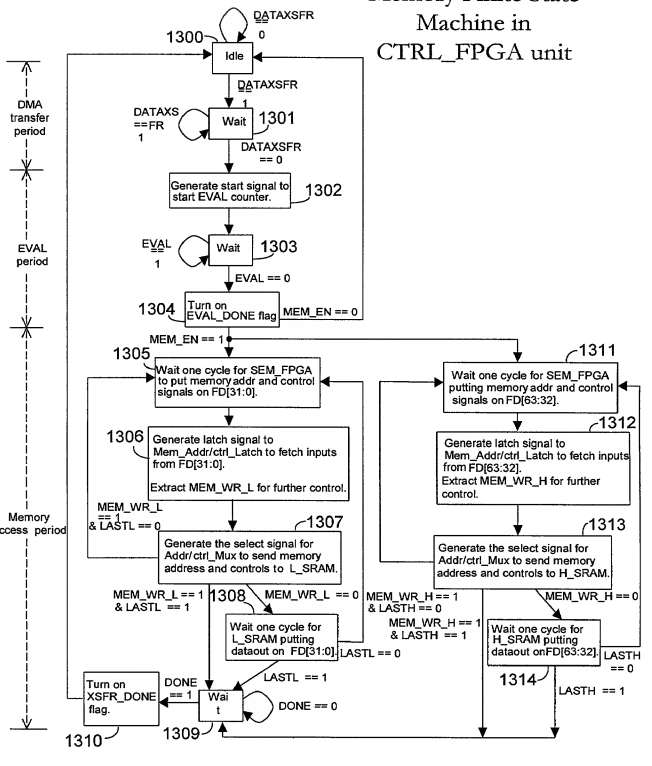


FIG. 58

EVALFSM - EVAL
Finite State Machine in
each FPGA logic device

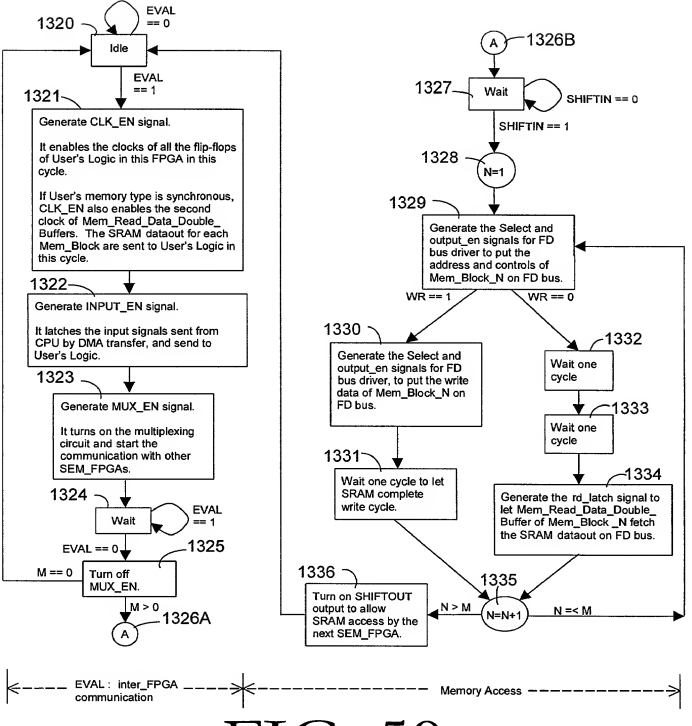


FIG. 59

MEMORY READ DATA DOUBLE BUFFER

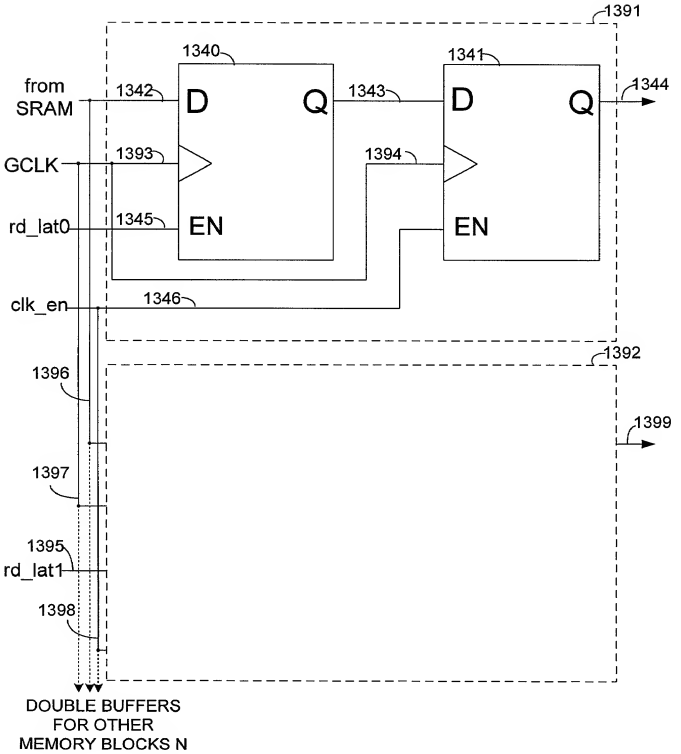


FIG. 60

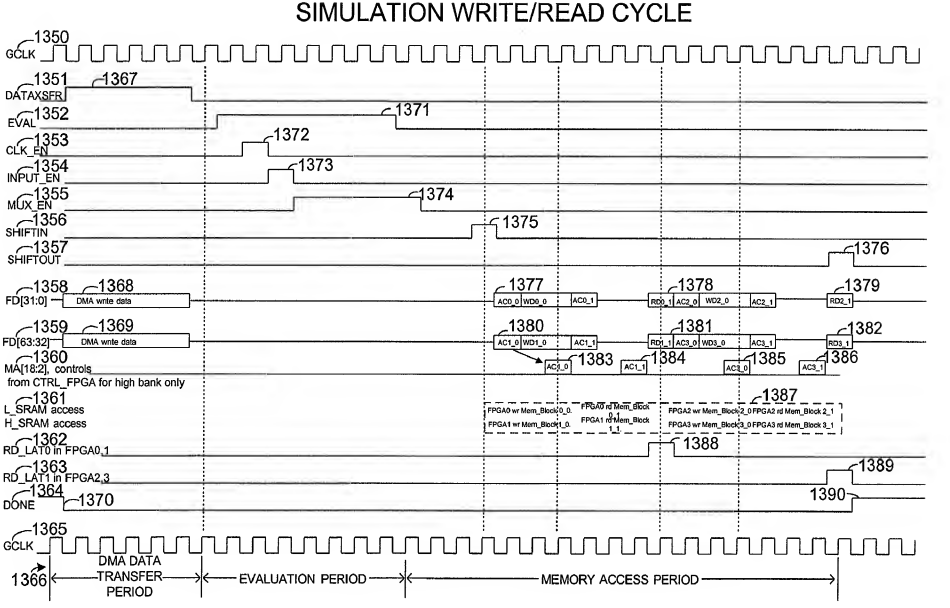

SIMULATION WRITE/READ CYCLE
1350
GCLK
1351
DATAXSFR
1367
1352
EVAL
1371
1353
CLK_EN
1372
1354
INPUT_EN
1373
1355
MUX_EN
1374
1356
SHIFTIN
1375
1357
SHIFTOUT
1376
1358
FD[31:0]
1368
DMA write data
1377
AC0_0
WD0_0
AC0_1
1378
RD0_1
AC2_0
WD2_0
AC2_1
RD2_1
1379
1359
FD[63:32]
1369
DMA write data
1380
AC1_0
WD1_0
AC1_1
1381
RD1_1
AC3_0
WD3_0
AC3_1
RD3_1
1382
1360
MA[18:2], controls
from CTRL_FPGA for high bank only
AC3_0
1383
AC1_1
1384
AC1_0
1385
AC3_1
1386
1361
L_SRAM access
H_SRAM access
1387
FPGA0 wr Mem_Block 0_0
FPGA1 wr Mem_Block 1_0
FPGA0 rd Mem_Block 1_1
FPGA1 rd Mem_Block 0_1
FPGA2 wr Mem_Block 2_0
FPGA3 wr Mem_Block 3_0
FPGA2 rd Mem_Block 2_1
FPGA3 rd Mem_Block 3_1
1362
RD_LAT0 in FPGA0,1
1388
1363
RD_LAT1 in FPGA2,3
1389
1364
DONE
1370
1390
1365
GCLK
1366
DMA DATA TRANSFER PERIOD
EVALUATION PERIOD
MEMORY ACCESS PERIOD

FIG. 61

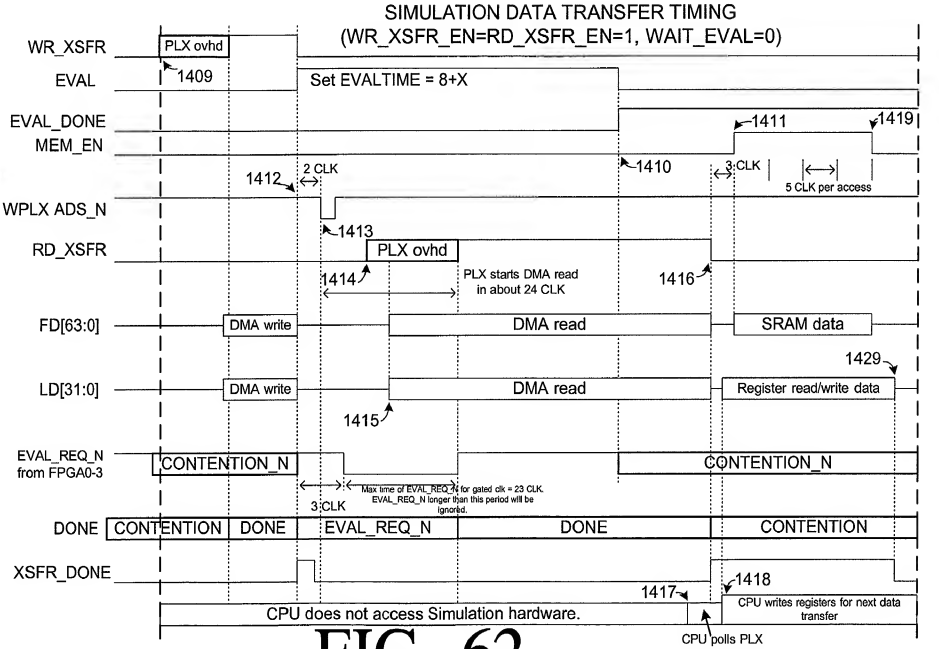

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)
WR_XSFR
PLX ovhd
EVAL
1409
Set EVALTIME = 8+X
EVAL_DONE
MEM_EN
1411
1419
1410
3 CLK
5 CLK per access
1412
2 CLK
WPLX ADS_N
1413
RD_XSFR
PLX ovhd
1414
PLX starts DMA read
in about 24 CLK
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1415
EVAL_REQ_N
from FPGA0-3
CONTENTION_N
CONTENTION_N
Max time of EVAL_REQ_N for gated clk = 23 CLK.
EVAL_REQ_N longer than this period will be
ignored.
3 CLK
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1417
1418
CPU writes registers for next data
transfer
CPU does not access Simulation hardware.
CPU polls PLX
interrupt register

FIG. 62

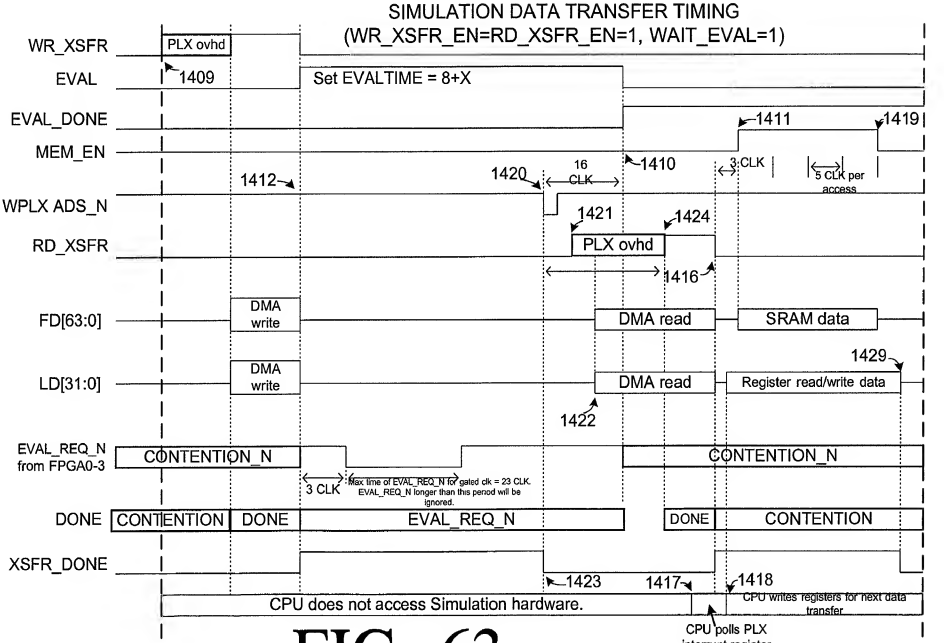


FIG. 63

Typical User Design of PCI Add-on Cards

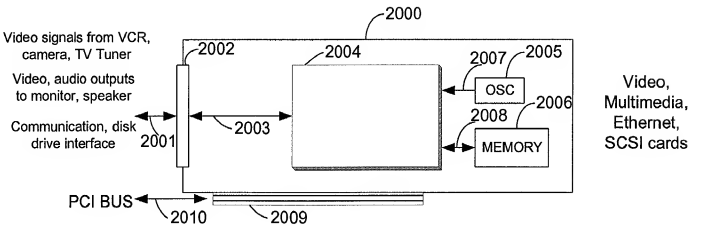


FIG. 64

Typical Hardware/Software Co-Verification

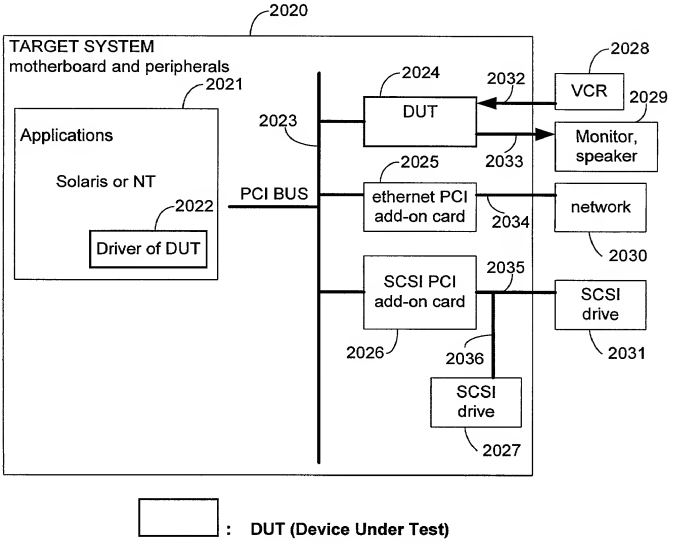
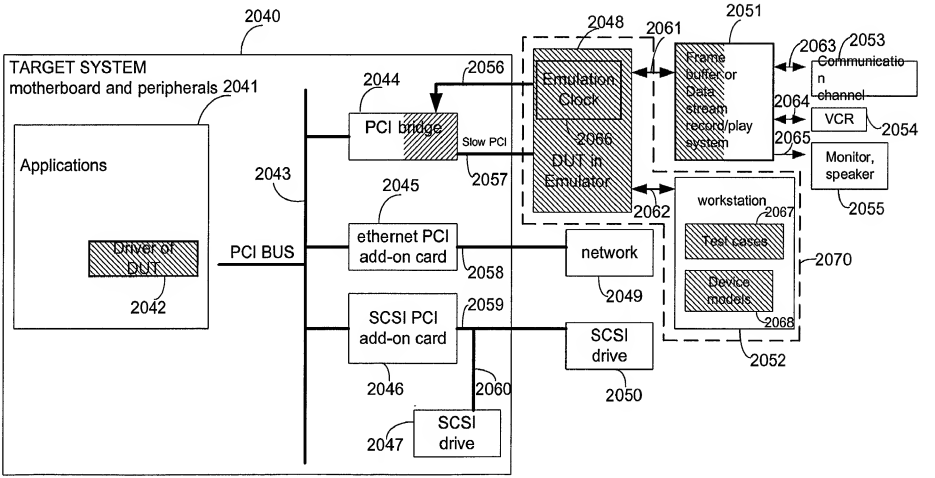


FIG. 65

Typical Co-Verification by Using Emulator



: running time at emulation speed

The rest of the target system is running at full speed.

FIG. 66

SIMULATION

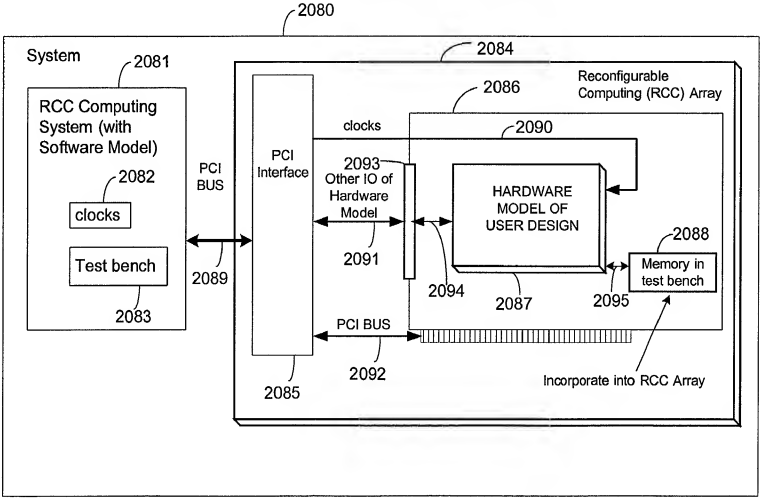


FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

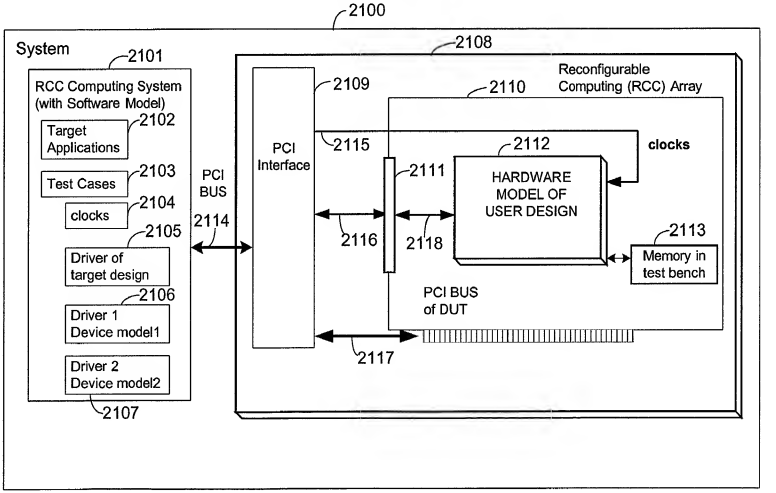


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

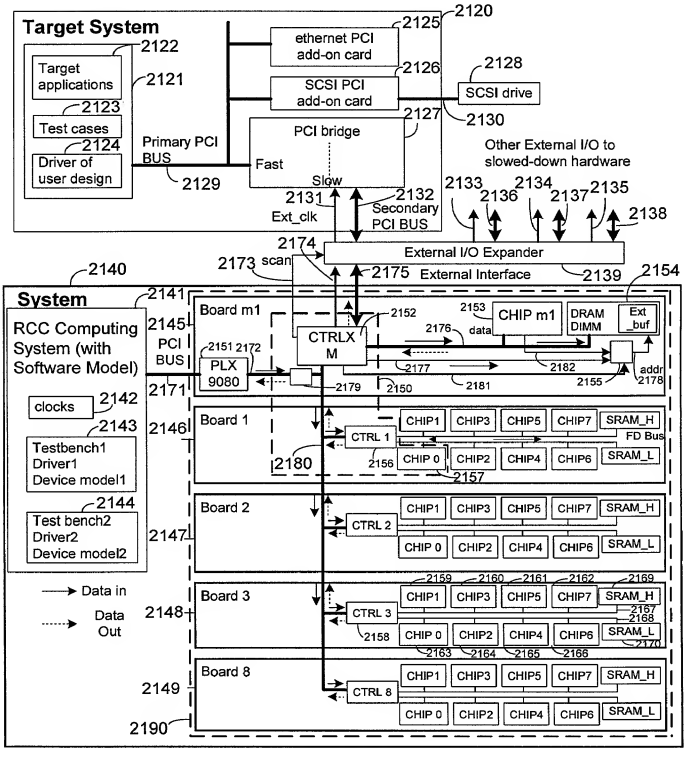


FIG. 69

CONTROL OF DATA-IN CYCLE

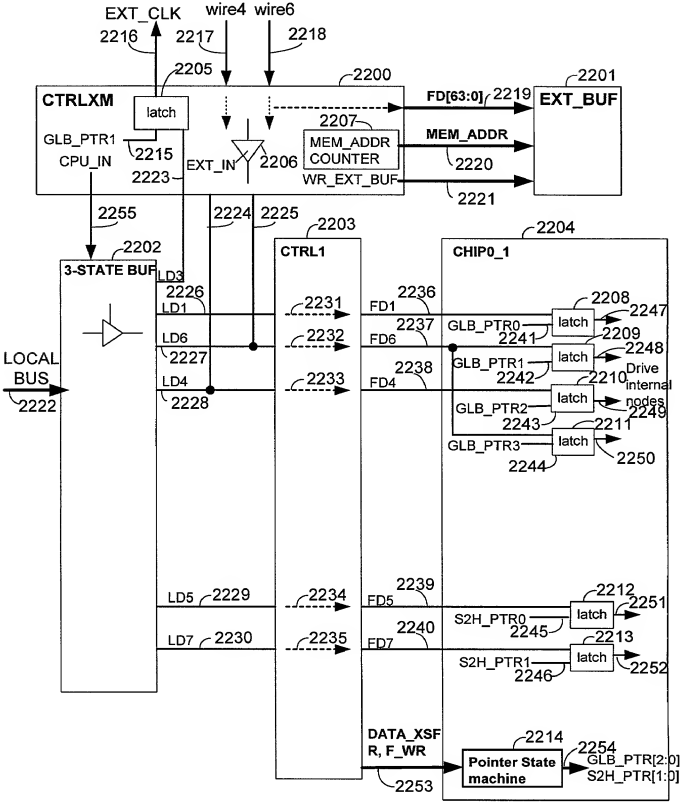


FIG. 70

CONTROL OF DATA-OUT CYCLE

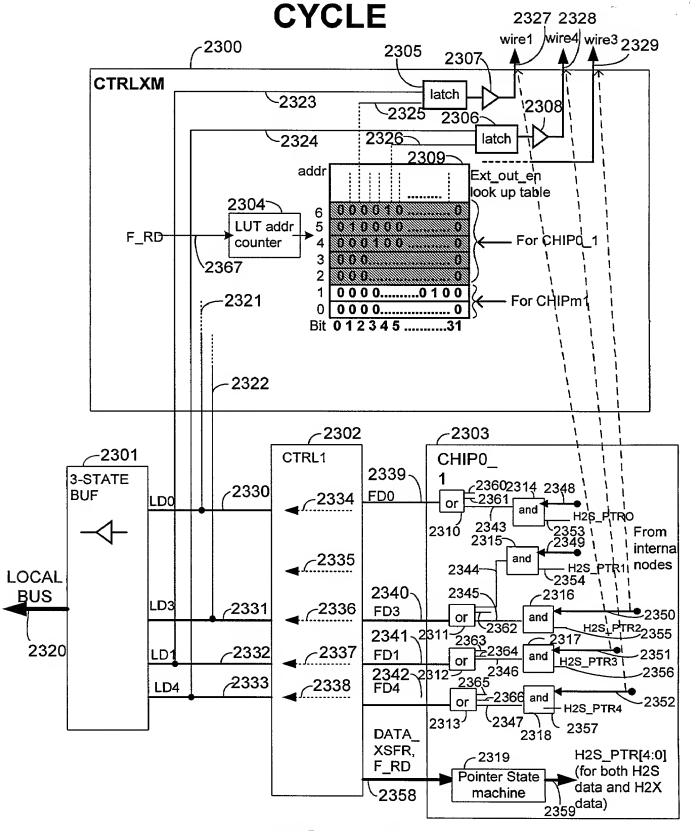


FIG. 71

CONTROL OF DATA-IN CYCLE

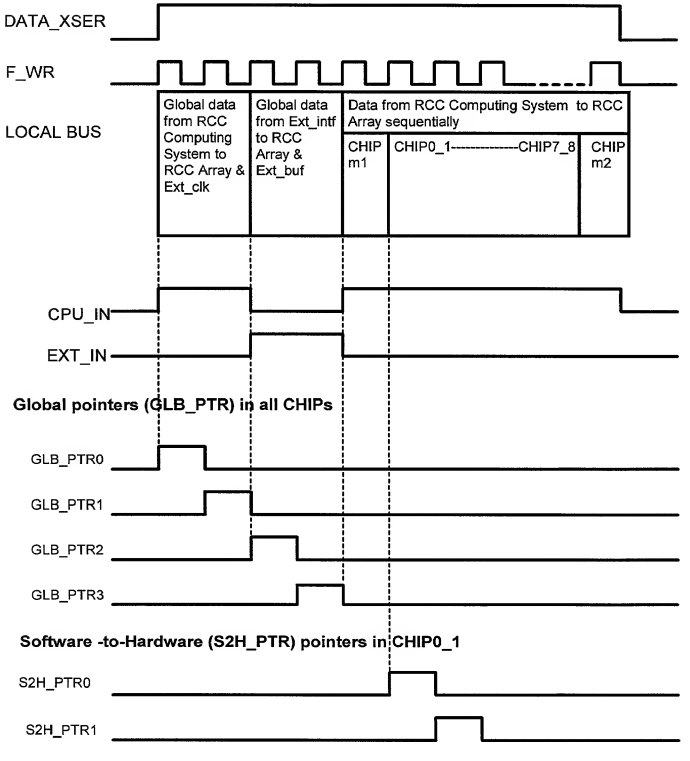


FIG. 72

CONTROL OF DATA-OUT CYCLE

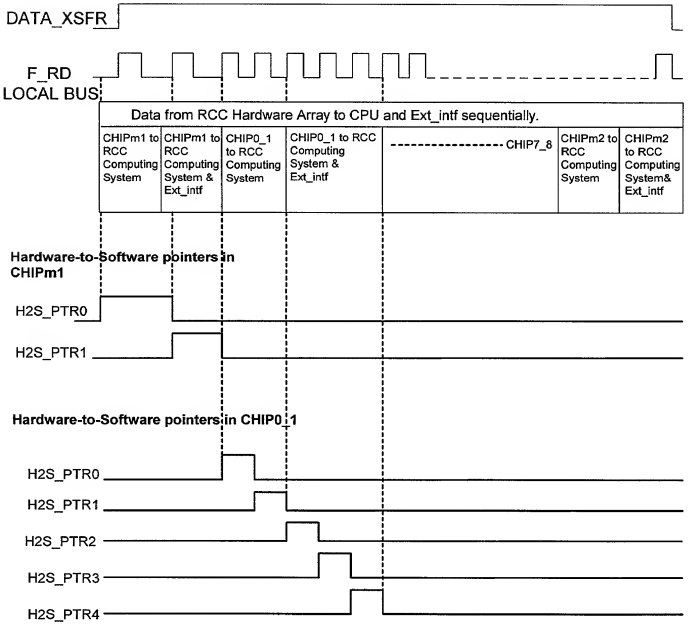


FIG. 73

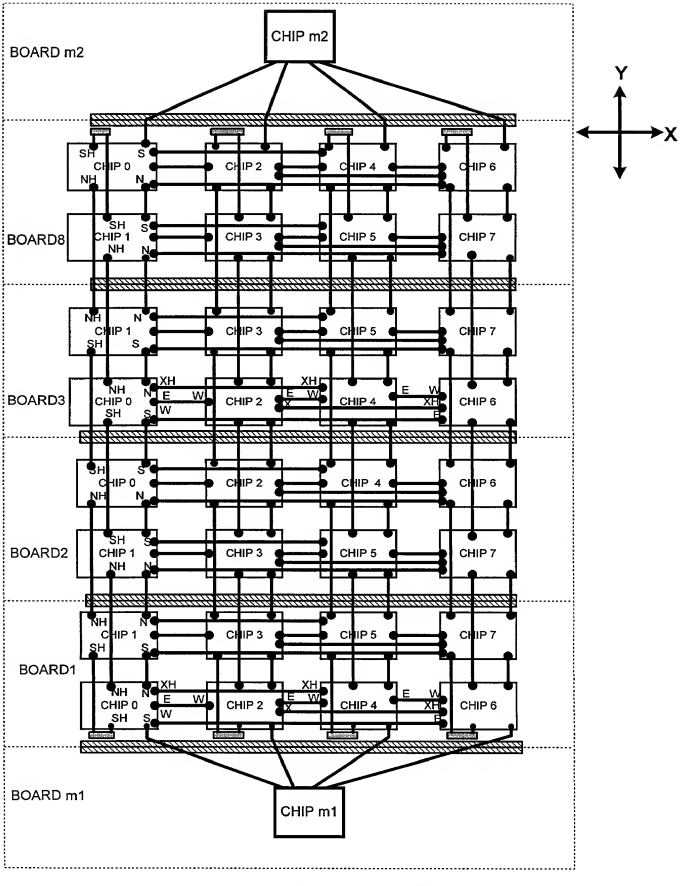


FIG. 74

SHIFT REGISTER

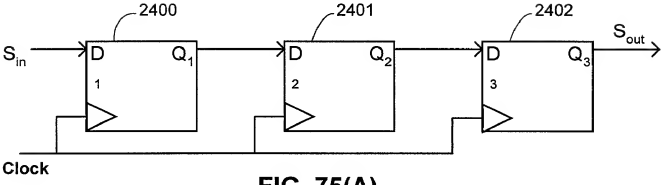


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

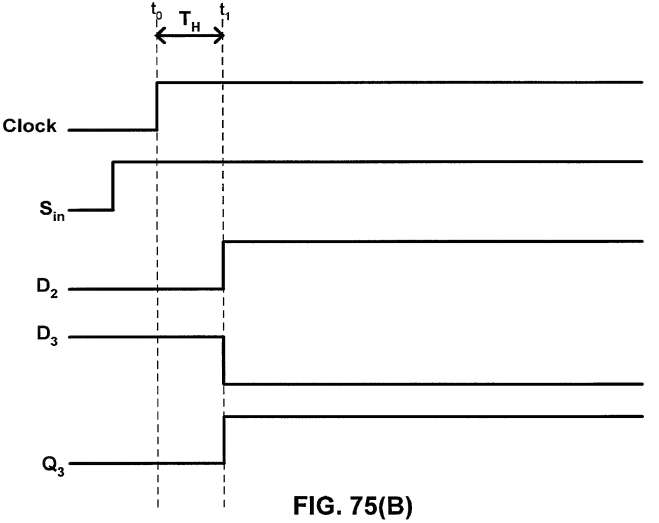


FIG. 75(B)

MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

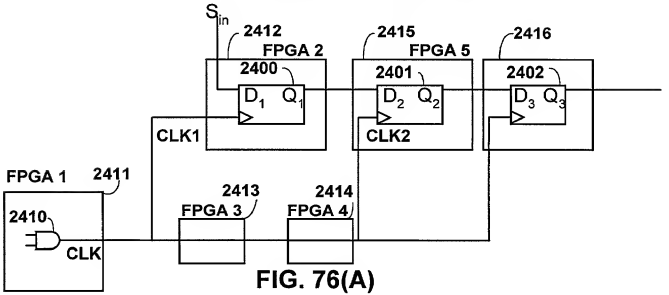


FIG. 76(A)

HOLD TIME VIOLATION BY LONG CLOCK SKEW

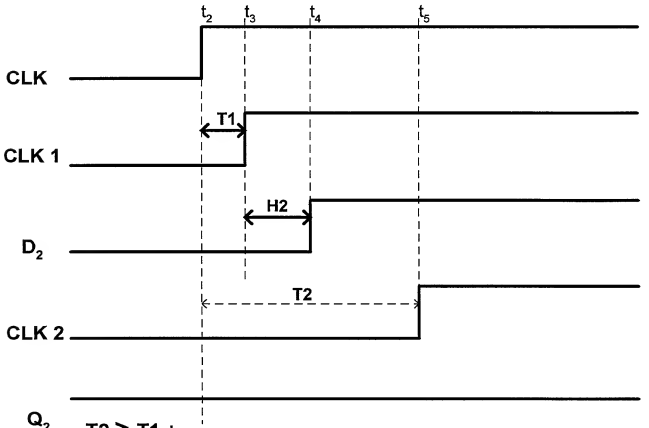


FIG. 76(B)

CLOCK GLITCH PROBLEM

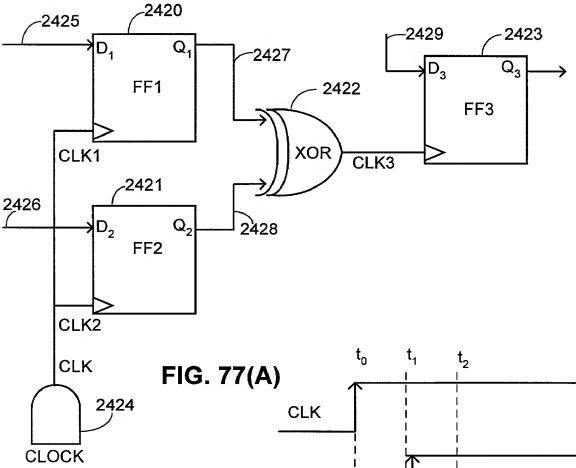


FIG. 77(A)

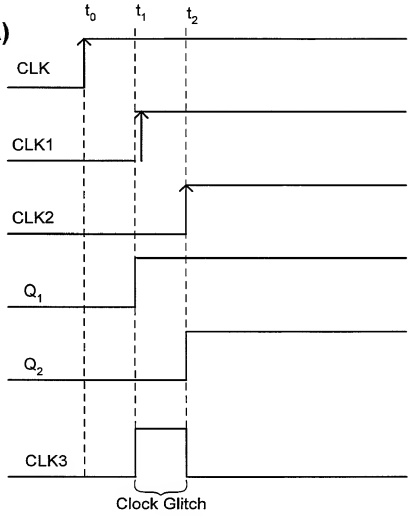
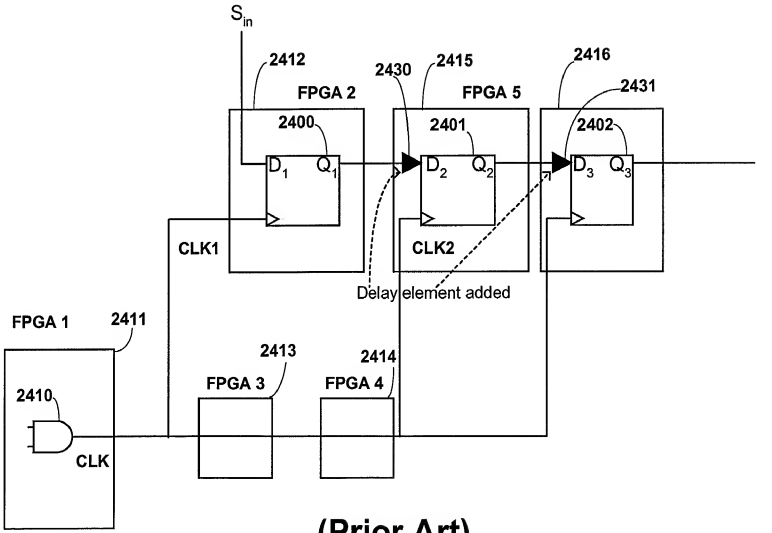


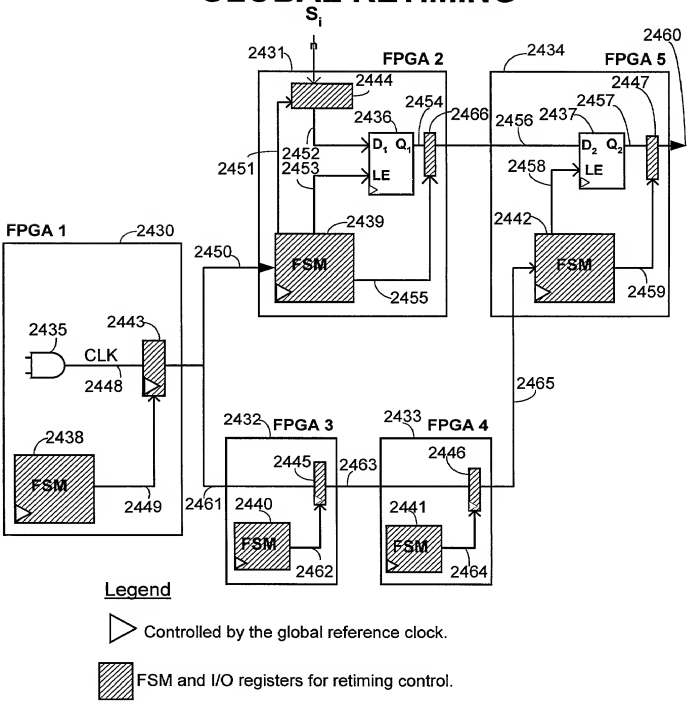
FIG. 77(B)

TIMING ADJUSTMENT BY ADDING DELAY



(Prior Art)

FIG. 78



(Prior Art)

FIG. 79

TIGF LATCH

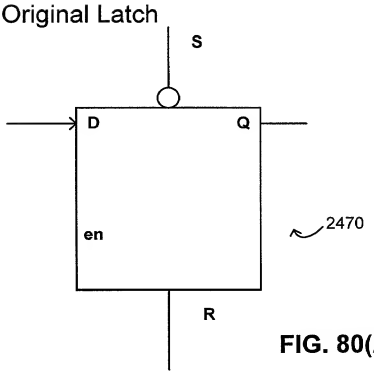


FIG. 80(A)

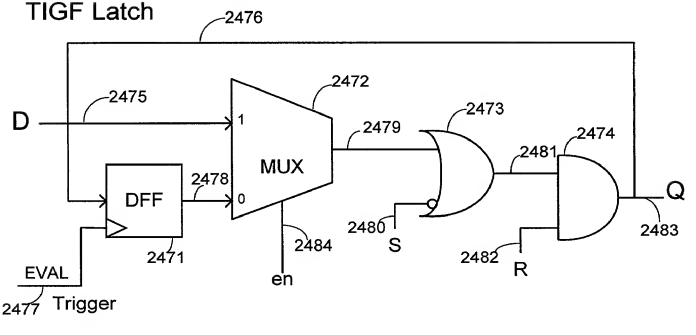


FIG. 80(B)

TIGF DFF

Original DFF

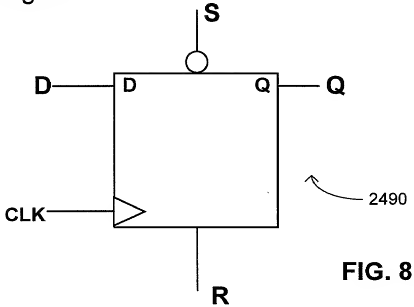


FIG. 81(A)

TIGF DFF and Edge Detector

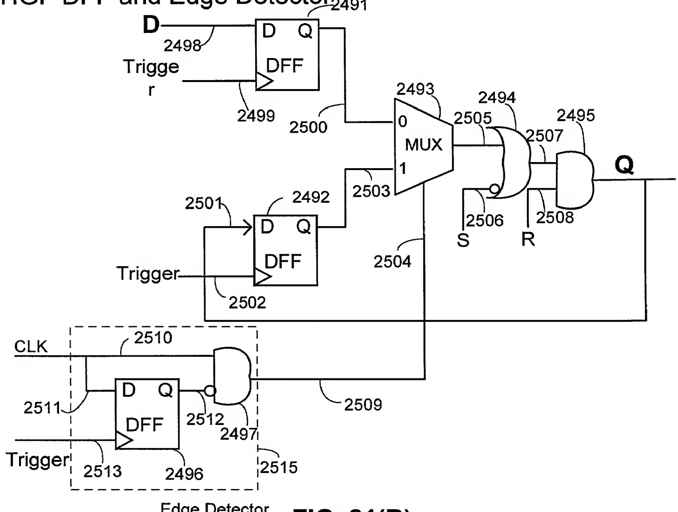


FIG. 81(B)

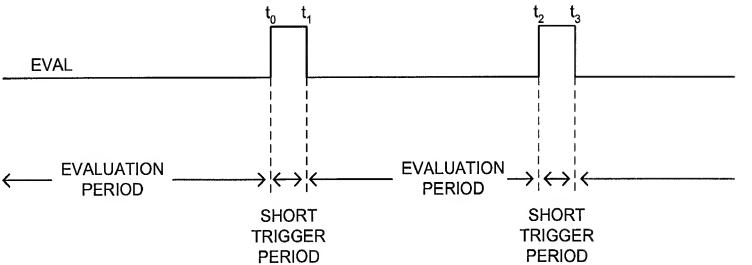


FIG. 82

RCC System

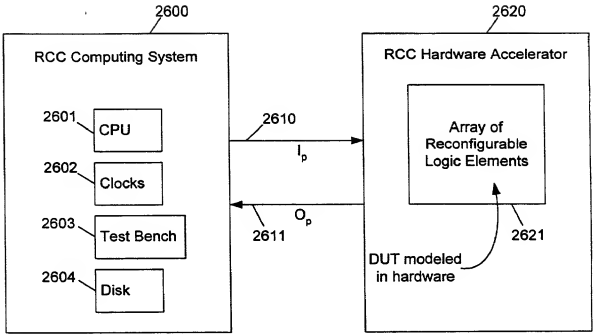


FIG. 83

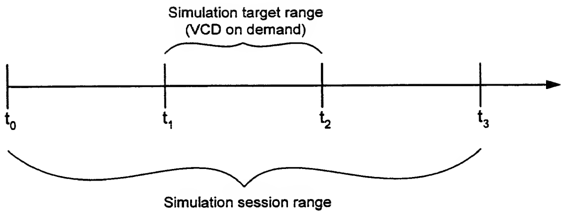

Simulation target range
(VCD on demand)
t0
t1
t2
t3
Simulation session range

FIG. 84

SINGLE-ROW FPGA PER BOARD

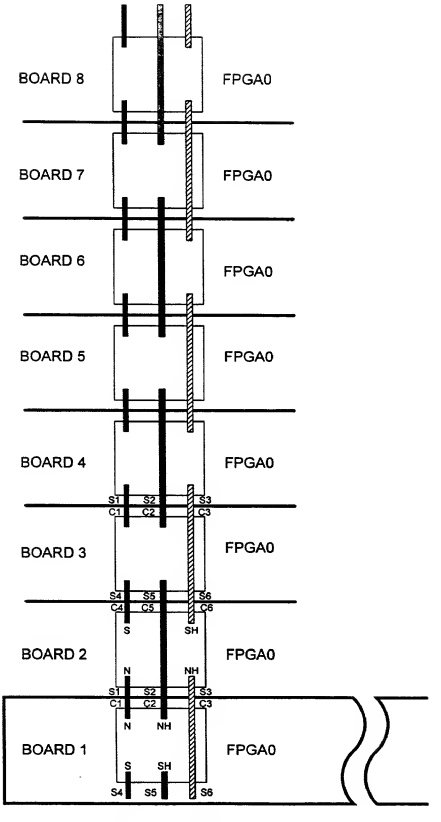


FIG. 85

TWO-ROW
FPGA PER
BOARD

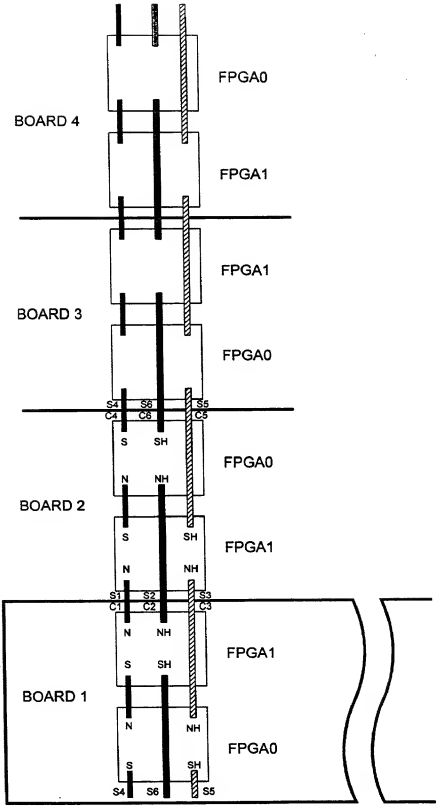


FIG. 86

THREE-ROW
FPGA PER
BOARD

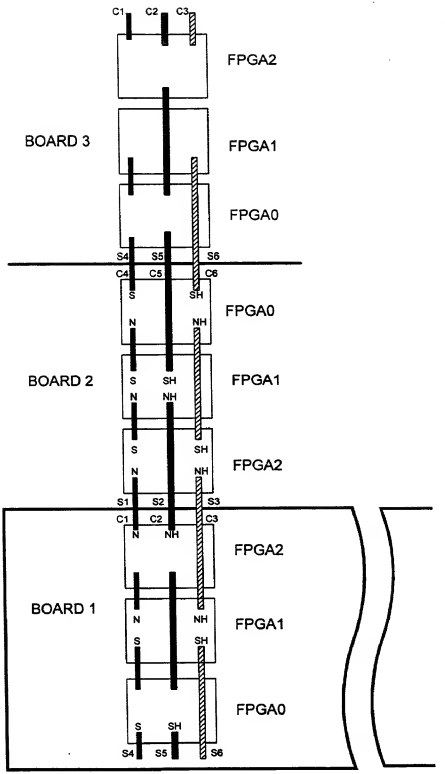


FIG. 87

FOUR-ROW FPGA PER BOARD

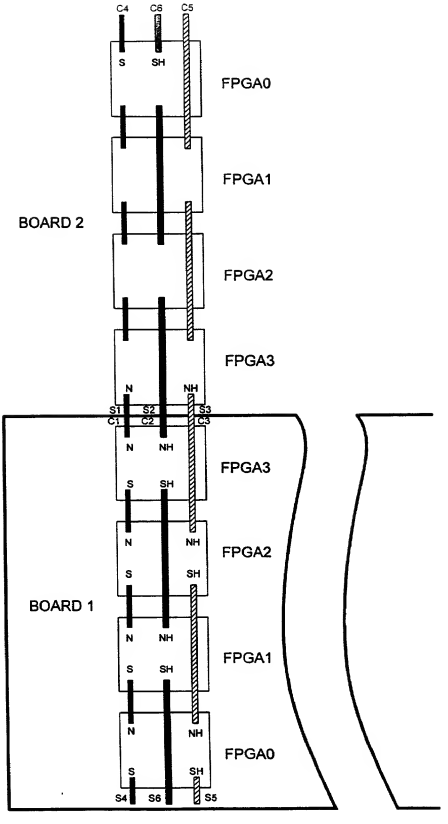


FIG. 88

INTERCONNECT FOR THREE-ROW PER BOARD

I/O Signals	Odd Board	Even Board	Common Board
	Connector-Group Pin-position	Connector-Group Pin-position	Connector-Group Pin-position
FPGA2_N	C1	S1	C1, S1
FPGA2_NH	C2	S3	C2, S3
FPGA1_NH	C3	S2	C3, S2
FPGA0_S	S4	C4	C4, S4
FPGA0_SH	S5	C6	C6, S5
FPGA1_SH	S6	C5	C5, S6

FIG. 89

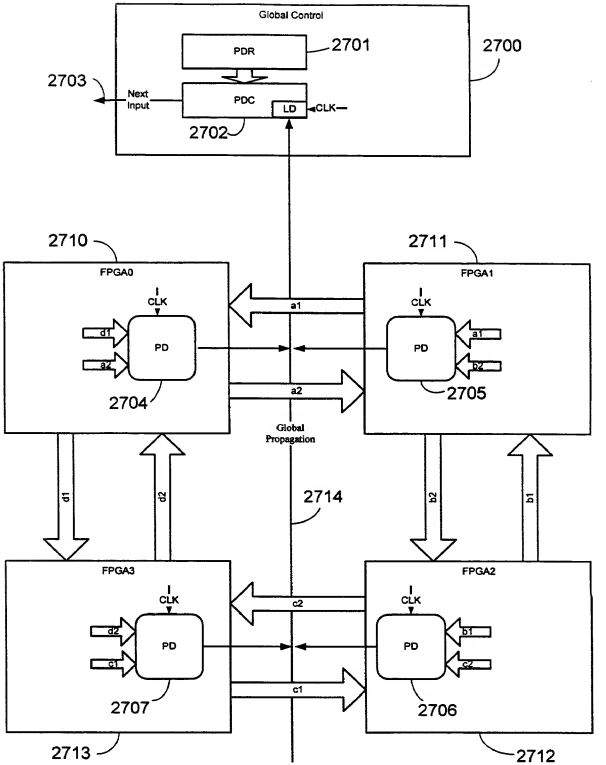

Global Control
PDR
2701
2700
2703
Next Input
PDC
LD
CLK
2702
2710
FPGA0
CLK
d1
d2
PD
2704
2711
FPGA1
CLK
a1
a2
PD
2705
Global Propagation
2714
FPGA3
CLK
c2
c1
PD
2707
FPGA2
CLK
b1
b2
PD
2706
2713
2712

FIG. 90

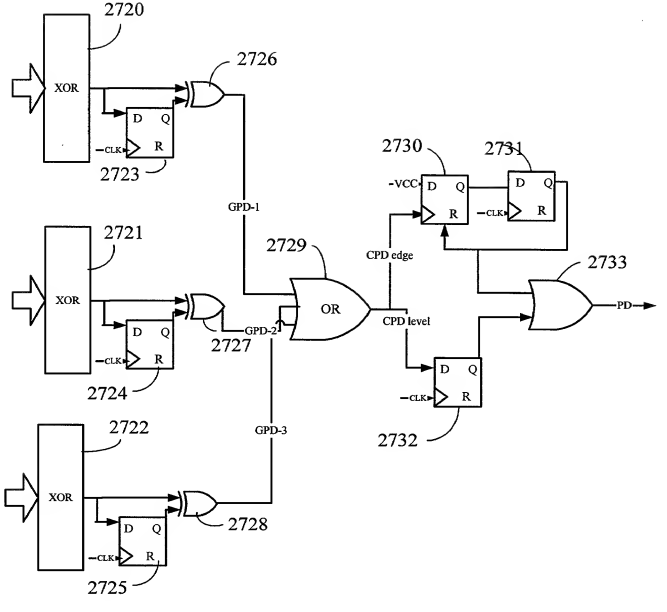


FIG. 91

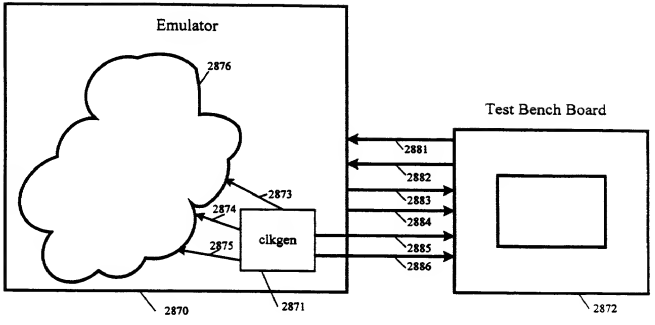

Emulator
2876
Test Bench Board
2881
2882
2883
2884
2885
2886
2873
2874
2875
clkgen
2870
2871
2872

FIG. 92

Clock Specification

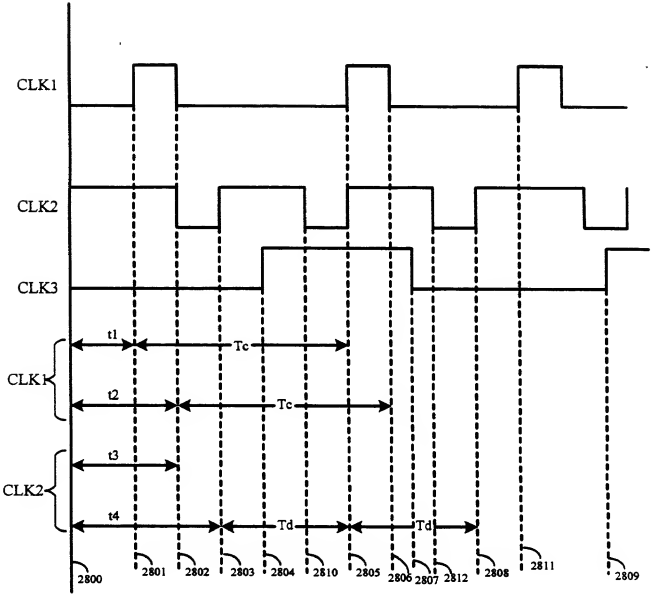


FIG. 93

Clock Generation Scheduler w/ Slices

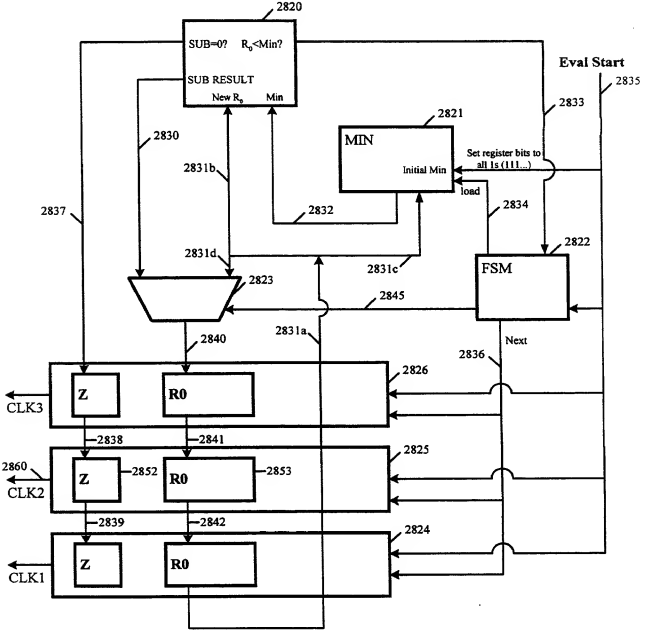


FIG. 94

Clock Generation Slice

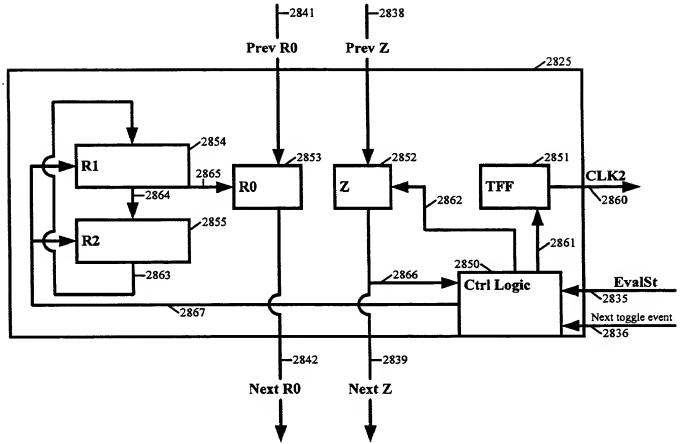


FIG. 95

Clock Generation Scheduler and Slices

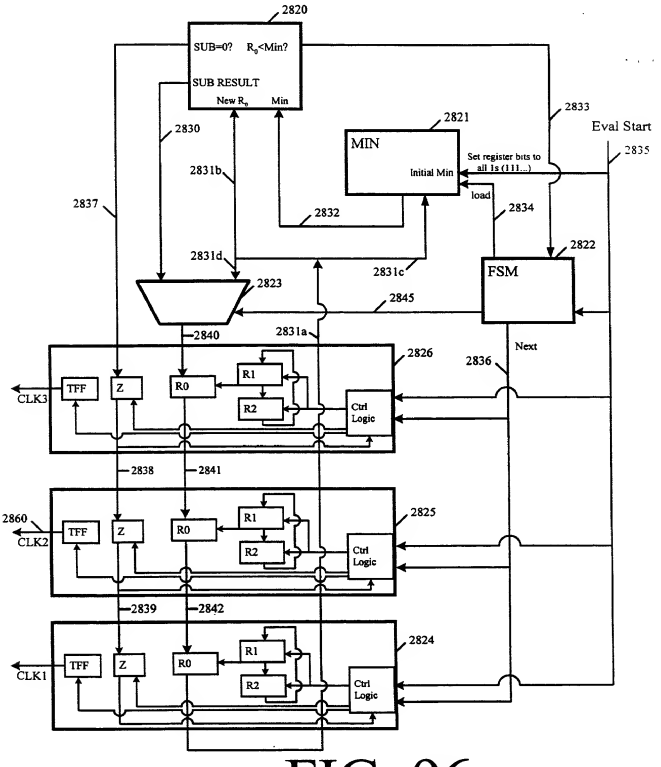


FIG. 96

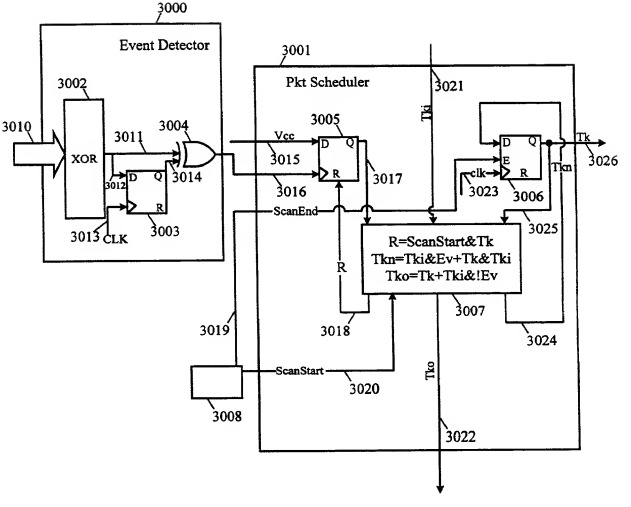


FIG. 97

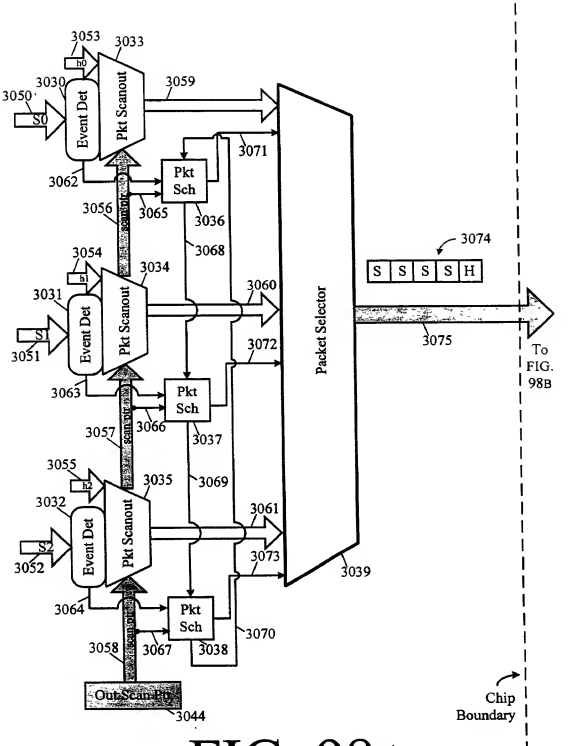


FIG. 98A

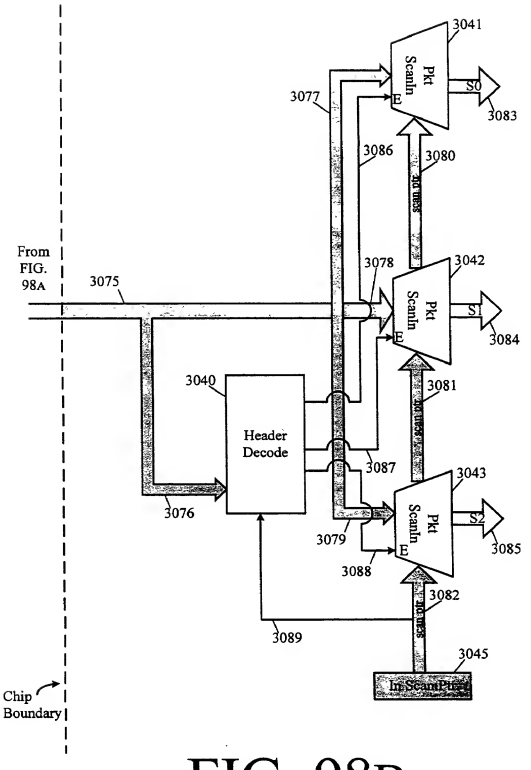

From FIG. 98A
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3040
Header Decode
3041
3042
3043
3045
Pkt ScanIn
SI
SO
E
400 bits
In ScanPort

FIG. 98B

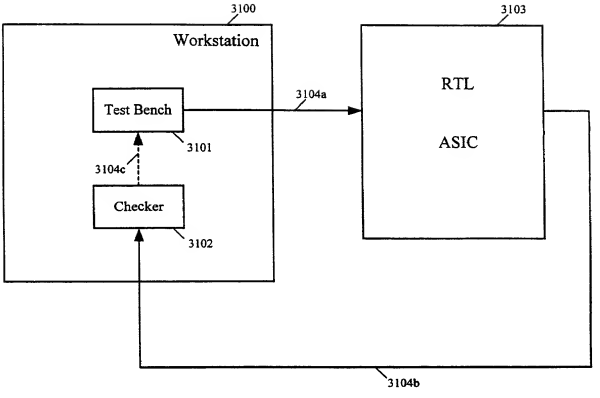

3100
Workstation
Test Bench
3101
3104c
Checker
3102
3104a
3103
RTL
ASIC
3104b

FIG. 99

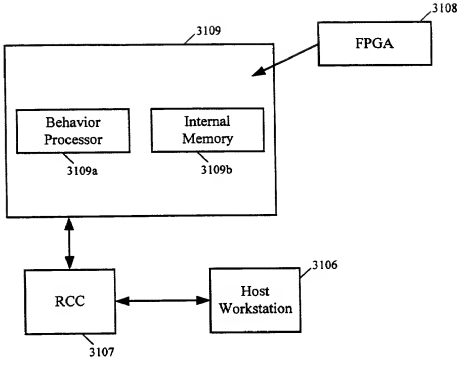

3108
FPGA
3109
Behavior Processor
3109a
Internal Memory
3109b
RCC
3107
Host Workstation
3106

FIG. 100

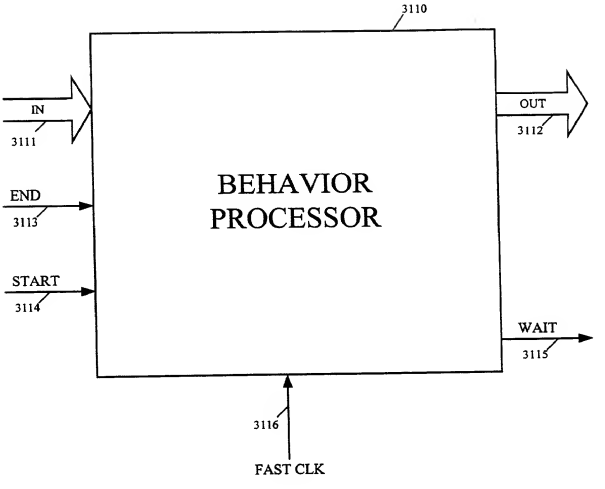

3110
IN
3111
END
3113
START
3114
BEHAVIOR
PROCESSOR
OUT
3112
WAIT
3115
3116
FAST CLK

FIG. 101

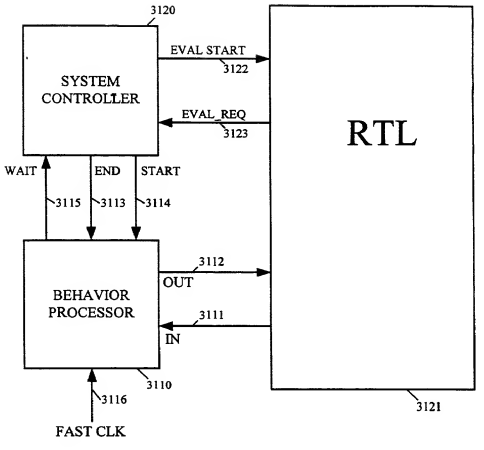

3120
SYSTEM CONTROLLER
EVAL START
3122
EVAL_REQ
3123
RTL
WAIT
END
START
3115
3113
3114
3112
OUT
BEHAVIOR PROCESSOR
3111
IN
3110
3116
FAST CLK
3121

FIG. 102

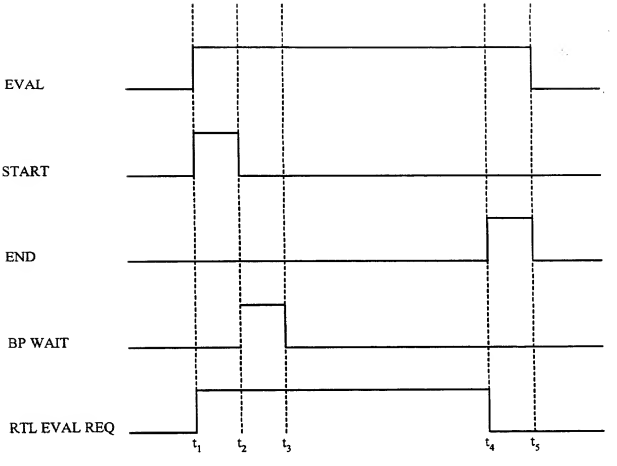


FIG. 103

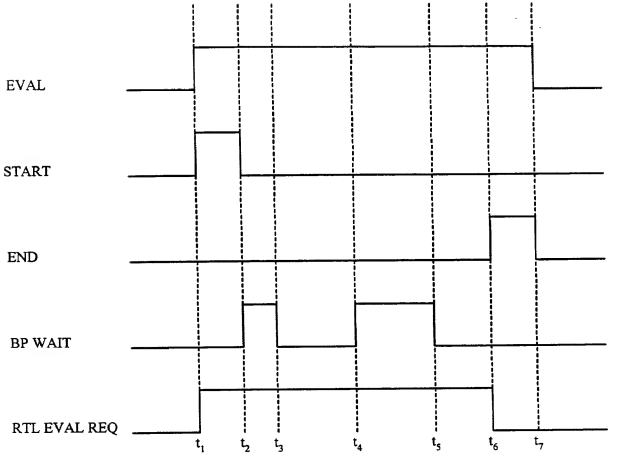


FIG. 104

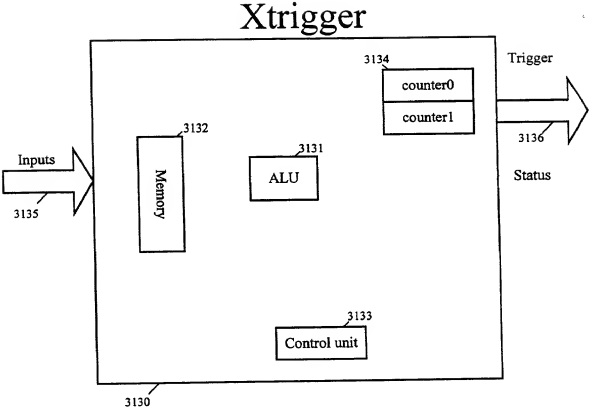

Xtrigger
3134
counter0
counter1
Trigger
3136
Status
3132
Memory
3131
ALU
Inputs
3135
3133
Control unit
3130

FIG. 105